NGSIM-VIDEO User's Manual

Publication No. #FHWA-HOP-07-009 **March 2007**

U.S. Department of Transportation
Federal Highway Administration

Notice

Quality Assurance Statement

Technical Report Documentation Page

1. Report No. FHWA-HOP-07-009	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle NGSIM-VIDEO User's Manual		5. Report Date March 2007 6. Performing Organization Code
7. Author(s) Lin Zhang, Vijay Kovvali, Nathan Clark, Douglas Sallman, and Vassili Alexiadis		8. Performing Organization Report No.
9. Performing Organization Name and Address Cambridge Systematics, Inc. 100 CambridgePark Drive, Suite 400 Cambridge, MA 02140		10. Work Unit No. 11. Contract or Grant No. DTFH61-02-C-00036
12. Sponsoring Agency Name and Address Department of Transportation Federal Highway Administration Office of Acquisition Management 400 Seventh Street S.W., Room 4410 Washington, DC 20590		13. Type of Report and Period Covered Final Report July 2003 – August 2006 14. Sponsoring Agency Code

15. Supplementary Notes
FHWA COTR: John Halkias, Office of Operations, and James Colyar, Office of Operations R&D

16. Abstract

The user manual for the Next Generation Vehicle Interaction and Detection Environment for Operations (NGSIM-VIDEO) software is presented in this document. NGSIM-VIDEO software extracts vehicle positions from video obtained from multiple cameras and translates it into vehicle tracking data. This manual covers the various steps required for developing accurate vehicle trajectories from video collection guidelines to trajectory data checking. The manual provides: 1) NGSIM-VIDEO installation steps; 2) guidelines to ensure collected videos meet the image requirements for NGSIM-VIDEO processing; 3) preprocessing of the video for NGSIM-VIDEO detection and tracking; 4) video processing through the NGSIM-VIDEO program to obtain vehicle trajectories; and 5) checking the database to ensure that the trajectories are accurate. Detailed instructions are provided in the manual for the preprocessing steps, consisting of image stabilization, camera calibration, and image rectification. Step-by-step instructions are provided to guide users in the installation of NGSIM-VIDEO and the database. Vehicle detection and tracking using NGSIM-VIDEO are described in detail. The report also describes the database used to store NGSIM-VIDEO output data and provides brief instructions on how to manipulate the database. Two real-world examples of using NGSIM-VIDEO, one for a freeway and the other for an arterial with signalized intersections, are provided to illustrate the application of NGSIM-VIDEO.

17. Key Words Next Generation Simulation, NGSIM, traffic simulation, vehicle trajectory, detection, tracking, image preprocessing.	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161.		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of Pages 82	22. Price

Form DOT F 1700.7 (8-72) **Reproduction of completed pages authorized**

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
^oF	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	^oC
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
^oC	Celsius	1.8C+32	Fahrenheit	^oF
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380.
(Revised March 2003)

Table of Contents

List of Tables

List of Figures

1.0 Overview

NGSIM-VIDEO, **N**ext **G**eneration **SIM**ulation - **V**ehicle **I**nteraction and **D**etection **E**nvironment for **O**perations, was developed to extract vehicle positions from video obtained from multiple cameras and translate it into vehicle tracking data. The data assist transportation engineers and researchers in the microscopic study of vehicle interactions in traffic conditions ranging from free flow to congested conditions.

NGSIM-VIDEO is an open source software application distributed by the Federal Highway Administration (FHWA) for use by the general public free of charge. It is a program developed by Cambridge Systematics, Inc. through the Next Generation Simulation (NGSIM) project. For more information, please visit http://ngsim.fhwa.dot.gov/.

1.1 What Is NGSIM-VIDEO?

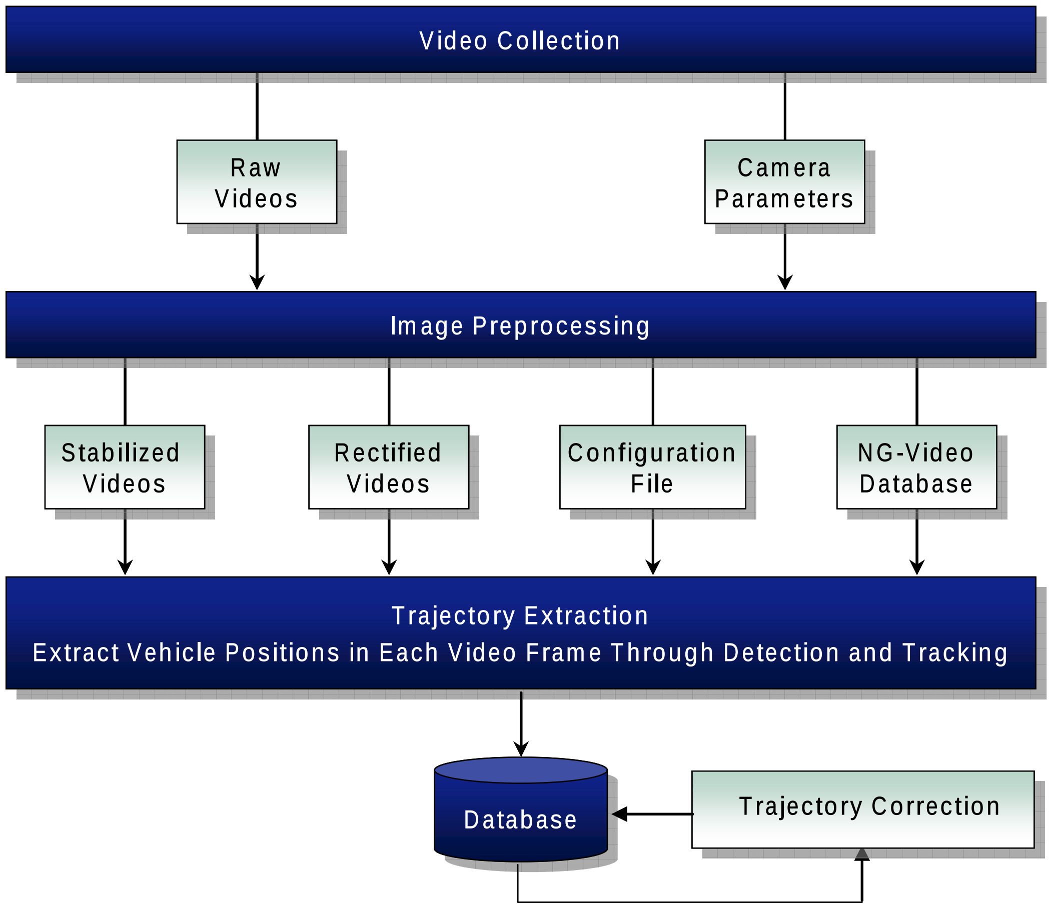
NGSIM-VIDEO detects and tracks each vehicle from one video frame to the next, and from one camera to the next using ortho-rectified images, creating a continuous trajectory for each vehicle in the time-space plane for the area of study. It also allows the user to manually detect vehicles that are not automatically picked up by NGSIM-VIDEO and correct individual trajectories that drift out of the true vehicle's path.

The overall vehicle tracking process consists of capturing video data on a roadway, preprocessing video images, and then extracting the vehicle trajectories from the video. The *Trajectory Extraction* phase involves using the NGSIM-VIDEO software to track vehicles and put the trajectory data into a database. The overall process is summarized in Figure 1.1.

1.2 Who Should Use NGSIM-VIDEO?

Transportation engineers and researchers, who are interested in extracting their own vehicle trajectory data for their study on traffic flow theory and simulation models, are encouraged to use NGSIM-VIDEO. Users should understand that using NGSIM-VIDEO requires significant efforts to collect suitable videos, preprocess the videos, prepare configuration files, understand MySQL database, and complete vehicle tracking. For example, previous efforts have required over 1,800 labor hours to compile 30 minutes of vehicle trajectory data on 6 lane-kilometers of highway. It is also required that the user should have moderate programming skills to post-process the data.

Figure 1.1 Trajectory Data Collection Process



1.3 NGSIM-VIDEO INPUTS

The primary inputs to NGSIM-VIDEO data processing are digital video files from one or more high-resolution cameras showing vehicle movements along a freeway or arterial corridor. NGSIM-VIDEO is designed to track vehicles using video images taken from a vantage point directly above the roadway, or as close to directly above the roadway as possible. Therefore, in order to capture any significant length of roadway, the video needs to be captured from a very tall structure located immediately adjacent to the roadway. Previous NGSIM-VIDEO processing efforts have utilized video footage captured from 30- and 36-story buildings.

The resolution of the video images must meet robust minimum requirements in order to provide sufficient clarity for the NGSIM-VIDEO program to identify individual vehicle characteristics. Ideally, the cameras used for data collection should be capable of capturing color digital images with 3 pixels per foot or higher resolution. Previous data processing efforts have shown that the resolution of the video footage is a critical determinant in the eventual effort required to process the vehicle trajectories.

Additionally, if multiple cameras are used to capture video, the camera placement must be carefully configured to provide sufficient overlap in views and synchronized to allow the video from the different views to be combined for simultaneous processing.

Prior to actually using the video images in the NGSIM-VIDEO program, significant pre-processing efforts are required to stabilize and rectify the images. This process requires the camera parameters that were used during the video collection and a high-resolution aerial photograph.

1.4 Resources Required To Use NGSIM-VIDEO

Using NGSIM-VIDEO requires a relatively high level of computing resources and additional software applications.

Computing Resources

NGSIM-VIDEO requires a fast processor to handle tracking of many vehicles simultaneously. It is recommended that the computer should at least have a CPU of 3.00 GHz, memory of 1.00 GB RAM, and a hard drive with 300 GB of free space. Also, dual monitors are typically required to simultaneously view the entire length or corridor being processed.

Software Resources

MATLAB® from the MathWorks, Inc. is necessary for camera calibration. ArcGIS and XMLSpy are recommended to prepare configuration files. MySQL, an open source database, is required for trajectory data storage. Another freeware, VirtualDub, is recommended for image editing.

Time Resources

Table 1.1 lists approximate estimates of time requirements for using NGSIM-VIDEO. Please note that raw video collection effort is not included in this estimate since it depends on many factors including location selections, permission acquirements, and interested time periods.

Table 1.1 Estimates of Time Requirements for Using NGSIM-VIDEO

Task	Estimated Time
Familiarity with NGSIM-VIDEO	16 – 24 hours
Familiarity with MySQL Database	12 – 24 hours
Video Stabilization	2 hours per hour of video
Camera Calibration	4 – 8 hours per camera
Image Correspondence	4 – 8 hours per camera
Camera World Boundary	4 – 8 hours per camera
Video rectification	4 – 8 hours per hour of video
Configuration File	16 – 24 hours
Tracking	Experienced NGSIM-VIDEO operators can effectively process video at the following approximate rates: • 40 – 100 seconds of tracking per work day for a 500-meter freeway section • 25 – 50 seconds of tracking per work day for a 500-meter arterial section

1.5 NGSIM-VIDEO OUTPUT

The output of NGSIM-VIDEO is a database that contains trajectories, including vehicle IDs, time, and vehicle positions. More detailed information about the output database format can be found in Section 5.0.

1.6 USER'S MANUAL STRUCTURE

This document provides instructions to the NGSIM-VIDEO users. The remainder of this report is divided into sections that describe the various steps to use NGSIM-VIDEO. The various sections include:

- **Section 2.0** provides more detailed information on video, hardware, and software requirements. Step-by-step instructions are also provided to guide a new user in the installation of NGSIM-VIDEO and MySQL database.
- **Section 3.0** describes video preprocessing in order to use NGSIM-VIDEO. The preprocessing procedure includes video stabilization, video rectification, and preparation of a configuration file.
- **Section 4.0** provides detailed instructions on how to use NGSIM-VIDEO to track vehicles.

- **Section 5.0** describes the database used to store NGSIM-VIDEO output data. This section helps users understand the database structure and provides brief instructions on how to manipulate the database.
- **Section 6.0** describes two real-world examples of using NGSIM-VIDEO, one for a freeway and the other for an arterial with signalized intersections.

2.0 Getting Started

This section describes the data guidelines for using NGSIM-VIDEO, including quality of videos, hardware, and software. Step-by-step instructions are also provided to guide a new user in the installation of NGSIM-VIDEO and MySQL database.

2.1 Video Guidelines

NGSIM-VIDEO is designed to track vehicles using video images taken from a vantage point directly above the roadway, or as close to directly above the roadway as possible. Therefore, in order to capture any significant length of roadway, the video needs to be captured from a very tall structure located immediately adjacent to the roadway. Previous NGSIM-VIDEO processing efforts have utilized video footage captured from 30 and 36 story buildings.

The resolution of the video images must meet high minimum requirements in order to provide sufficient clarity for the NGSIM-VIDEO program to identify individual vehicle characteristics. The ideal resolution should be 3 pixels per foot or higher, and the minimum resolution should be 1 pixel per foot. In order to meet this specification for anything other than a trivial length of roadway, multiple synchronized cameras must be used to simultaneously capture various sections of the roadway. The resulting videos are a set of *MPEG-4* standard Audio Video Interleave (AVI) files, compressed using the *XVID* codec at an *8,000 Kb* compression rate.

Other camera interface guidelines are listed in Table 2.1. These guidelines cover issues, such as camera placement, camera synchronization, raw video formats, and raw video resolution. In order for NGSIM-VIDEO to function correctly, these guidelines are recommended to be met in full. Note that trajectory extraction may still be performed without strictly meeting some of these guidelines, but the tracking cost might be significantly higher as much greater manual intervention will be required. Among all these guidelines, resolution has proven to be one of the most important factors as higher resolution video greatly enhances the ability of NGSIM-VIDEO to automatically complete trajectory extraction.

2.2 Hardware

The image processing capabilities of NGSIM-VIDEO require robust computer hardware to function efficiently. Table 2.2 presents the computer hardware and operating system which NGSIM-VIDEO was specifically configured. This is a recommended hardware and operating system configuration, but does not represent the minimum required.

Figure 2.1 Relationship between Camera Location (on Top of Building) and Observed Highway

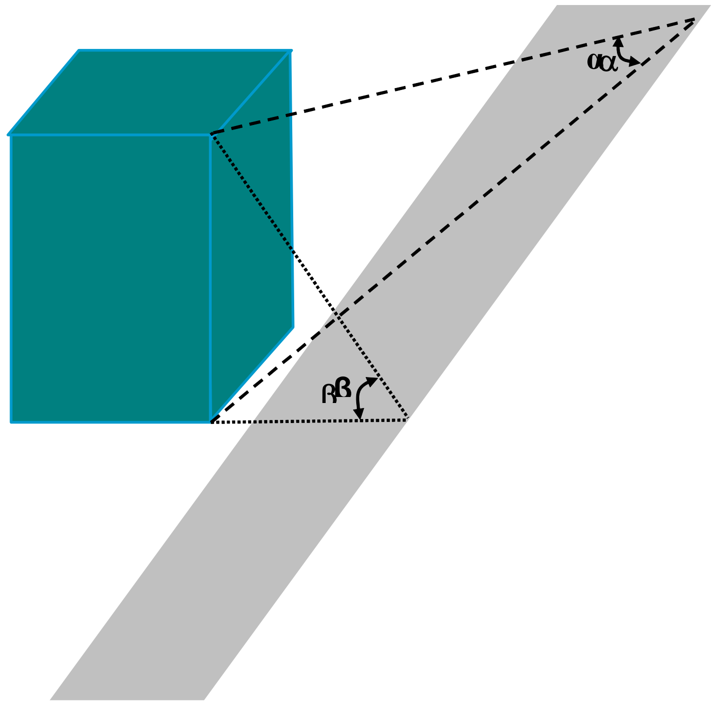


Table 2.1 Camera Interface Guidelines

Unit	Tolerance	Comments
90° resolution	3 pixel per foot or higher	Based on previous NGSIM tracking efforts, resolution is one of the most critical factors for smooth tracking.
α resolution	1 pixel per foot	Lower resolution means lower yields.
Minimum α	11°	Based on previous NGSIM tracking efforts, it is computationally (and physically) impossible to extract at a lower angle.
Minimum **β**	67°	Down to 50° may be possible physically, but it is likely to significantly decrease tracking efficiency as increasing rectification creates more image information loss. Less than 50° is not physically possible, as trucks will occlude more than half of car images.
Recommended minimum bit rate for compression	8 Mb/s	It is based on previous NGSIM experience and other research. This only applies up to 640x480 digital camera feeds. Higher rates apply for larger feeds.

Unit	Tolerance	Comments
Minimum frame rate	10 – 30 Hz	Depends on the desired detail of the output trajectory data. A multiple of 10 input frequencies (30 Hz on any typical camera) is suggested. If 10 frames/second is desired, users can use image pre-processing tools to process every third frame to achieve, as described in video editing in Section 3.0.
Maximum camera synchronization error	0.002 second	Camera synchronization within 0.002 seconds is highly recommended.
Minimum camera overlap	80 – 100 ft	If multiple cameras are used, minimum camera overlap should be observed. Even though NGSIM-VIDEO works well with rectified videos without overlaps, it would be easier to rectify videos if the minimum camera overlap is observed.
Maximum camera vibration	2 pixels/Hz	
Maximum effective occlusion length	40 – 50 ft	This is the distance of actual roadway occluded by trees or overpasses (which varies by viewpoint angle).
Minimum distance between occlusions (of any length)	80 – 100 ft	More frequent occlusions will prevent tracking vehicles effectively between occlusions. This requirement is somewhat flexible and depends greatly on the study area.
Intrinsic zoom	–	Cameras must have intrinsic zoom capability. Cameras must be calibrated at the exact zoom used for calibration.
Video format	MPEG4 w/XVID compression	Previous efforts have used this format. No other formats have been tested.
Video type	Color	It is recommended that all videos are shot in digital color, preferably true color.
Lighting	Daylight	It is recommended that videos are shot during daylight hours.

Table 2.2 System Requirements for Microsoft Windows-Based Computers

Processor	Multimedia PC Intel® Xeon™ CPU 3.00 GHz or faster processor
Memory	1.00 GB of RAM required
Hard Disk	300 GB of free hard-disk space
Display	Plug and Play Dual Monitors capable of displaying 1,600 x 1,200 resolution, 32-bit color
Operating System	Windows XP Professional Version 2002, SP2
Video Card	NVIDIA Quadro PCI-E Series video card or compatible video graphics adapter
Input Device	Microsoft mouse or compatible pointing device

2.3 SOFTWARE

Besides NGSIM-VIDEO, there are several other software applications which are recommended for extracting trajectory data. Table 2.3 presents the software applications which were used by the NGSIM team. Other equivalent software packages may be used as alternatives.

- MATLAB[1] for camera calibration and data processing.
- An open source MATLAB toolbox developed by Jean-Yves Bouguet[2] for camera calibration.
- ArcGIS[3] for acquiring image correspondence between video image and world-coordinate GIS map. It is also used for extracting zone boundaries, direction polygons, and directional points.
- XMLSpy[4] for creating schema and configuration files in xml format.
- SteadyHand[5] for video stabilization.
- MySQL,[6] an open source database, for trajectory data storage and query.

[1] http://www.mathworks.com/.

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[3] http://www.esri.com/software/arcgis.

[4] http://www.altova.com/.

[5] http://www.dynapel.com.

[6] http://www.mysql.com.

- VirtualDub[7] for image editing.
- A custom C++ application developed by the NGSIM team for image rectification.

2.4 INSTALLING NGSIM-VIDEO

Double-click the file "setup.exe." This may also be done by right-clicking on the file and then selecting *Open*. Follow the installation wizard, which is similar to installing most other Windows-based applications. The default folder for installing the program is C:\Program Files\NGSIM-VIDEO, but it may be changed to a preferred folder. The complete installation process could take several minutes, depending on the speed of the computer.

2.5 INSTALLING MYSQL AND NGSIM-VIDEO DATABASE

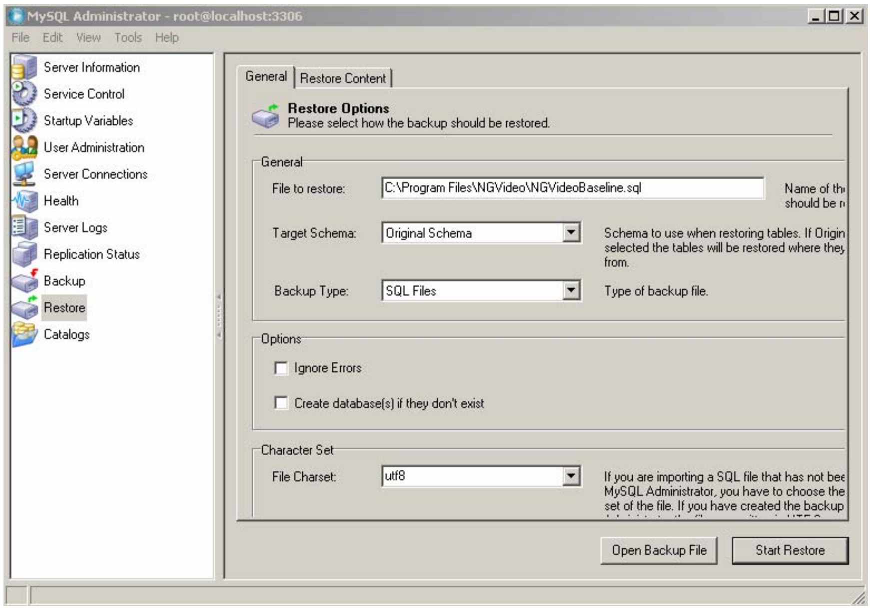
In order to use NGSIM-VIDEO, MySQL database needs to be installed and configured. The NGSIM-VIDEO installer will not install this by default. MySQL is an open source database and more information can be found on http://www.mysql.com/. The version of MySQL Server used during development was 4.1.8. It is not believed that a later version would pose any problem for NGSIM-VIDEO.

Both MySQL Server and MySQL Administrator have to be installed by following the MySQL installation procedure. It is critical that the user remember the password for "root" which is created when installing MySQL.

The installer provides a MySQL file used to create an empty database for NGSIM-VIDEO. This file is provided in the NGSIM-VIDEO runtime directory and is called "NGSIM-VIDEOBaseline.sql." MySQL Administrator may be used to create the database, as shown in Figure 2.2.

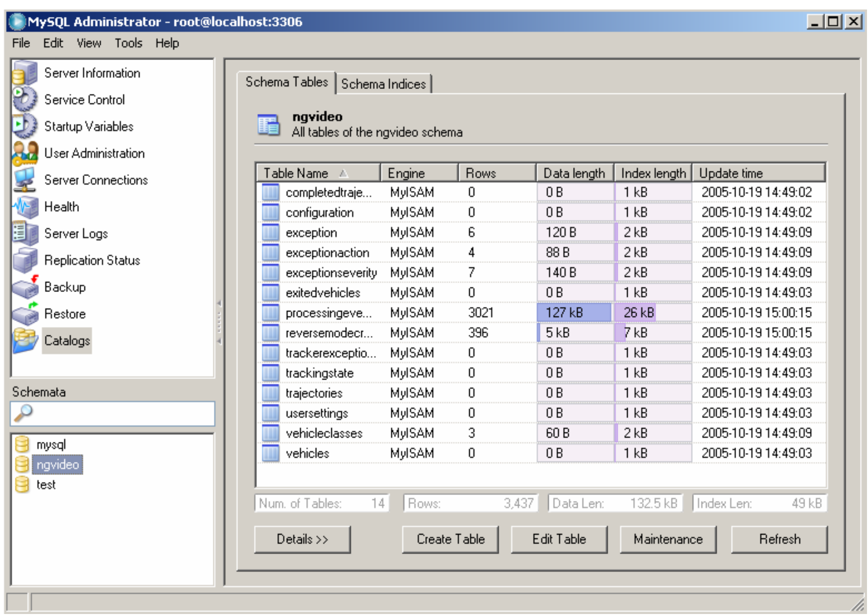
[7] http://www.virtualdub.org.

Figure 2.2 Create Database Using MySQL Administrator



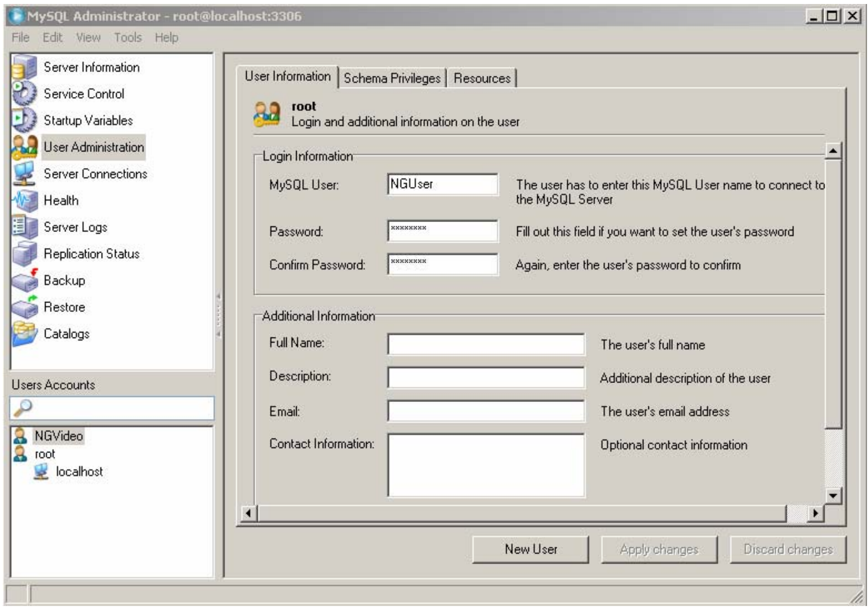
Users can verify if the empty database file NGSIM-VIDEOBaseline.sql has been installed successfully using MySQL Administrator, as shown in Figure 2.3.

Figure 2.3 Successful Creation of NGSIM-VIDEO Baseline Database



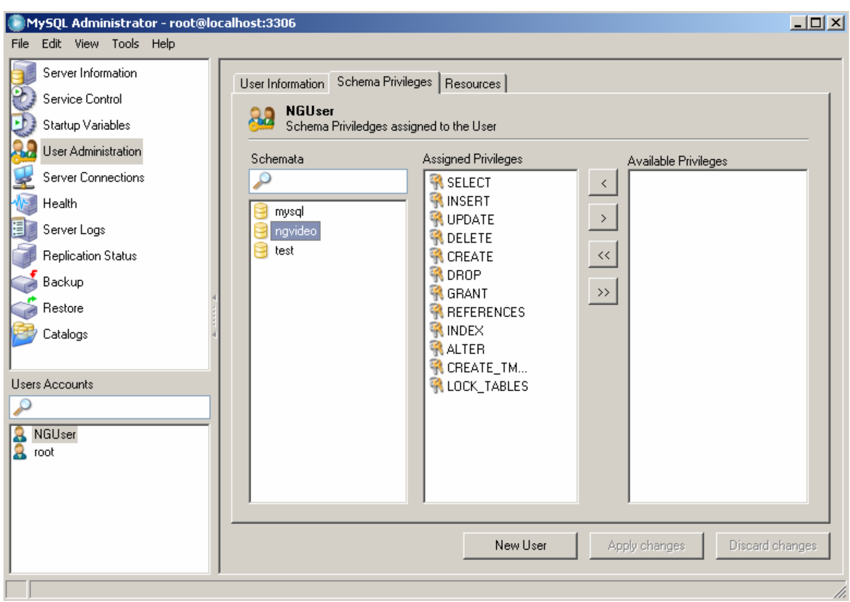
It is recommended that the user creates a new user account in MySQL and dedicate it to NGSIM-VIDEO. To do this, open MySQL Administrator, go to User Administration, and add a new user name and create a password. For example, a new user "NGUser" is created as shown in Figure 2.4. Note that all fields will be saved only when the "Apply Changes" button is clicked.

Figure 2.4 Add a New User Account in MySQL



In Schema Privileges tab of User Administration in MySQL Administrator, assign all privileges of schema "NGSIM-VIDEO" to the user account dedicated to NGSIM-VIDEO, similar to the example shown in Figure 2.5.

Figure 2.5 Assign Schema Privileges



3.0 Video Preprocessing

The video preprocessing phase formats the raw video files obtained from the field video collection into an input format necessary for NGSIM-VIDEO. It also supplies NGSIM-VIDEO with a set of parameters required to extract vehicle trajectories from videos.

The input to the video preprocessing phase is the raw video files from one or more cameras. As stated in the guidelines in Section 2.0, the raw videos are recommended to be a set of MPEG-4 standard AVI files, compressed using the XVID codec at an 8,000 Kb compression rate.

This section describes a series of manipulations to turn the video inputs into a set of outputs for use by NGSIM-VIDEO. The tools and procedures described in this section are:

- Edit and combine videos from multiple files into a single file for each camera or break monolithic videos into smaller files;
- Stabilize video from unstable camera platforms;
- Match the pixels in each video to real world coordinates through image correspondence;
- Rectify videos to provide accurate vehicle positions; and
- Establish configuration settings for the study area, including detection zones, drop zones, occlusion zones, and direction polygons.

By following these steps, the input files will be transformed into the following outputs:

- Stabilized, **original video files**. These will be a set of *MPEG-4* standard *AVI* files, compressed using the *XVID* codec at an *8,000 Kb* compression rate.
- Stabilized, **rectified video files**. These will be a set of *MPEG-4* standard *AVI* files, compressed using the *XVID* codec at an *8,000 Kb* compression rate. The video frames from adjacent cameras are mosaicked seamlessly.
- A **configuration file** in XML format, containing a set of configuration parameters. These include transformation matrices and other inputs that NGSIM-VIDEO requires for detection and tracking of vehicles.

These outputs will then be used as inputs by NGSIM-VIDEO to extract vehicle trajectories. The overall process followed to generate these outputs from the *Video Preprocessing* phase is shown in Table 3.1 and the details of this process are described in the remaining part of this section.

Table 3.1 Video Preprocessing Phase Overview

Description						Video Output		XML Output				
Step/Substep		Input	Tool	Intermediate Output	Optional?	Original	Rectified	Transform Matrices	Input Parameters	Zoning	Direction Polygons and Points	Configuration File
Video Editing		• Raw video	• VirtualDub Open Source application	Modified raw video	Yes	×						
Stabilization		• Raw video	• SteadyHand Commercial Off-the-Shelf (COTS) application	Stabilized video	Yes	×						
Video Rectification Process	Camera Calibration	• Calibration pattern frames • Zoom settings	• MATLAB with Open Source MATLAB toolbox	Camera intrinsic parameters	No							
	Image Correspondence	• Stabilized video frame • Ortho-rectified, geo-referenced aerial image	• ArcGIS COTS application • Custom C++ application	Image correspondence points	No							
	World Boundary	• Ortho-rectified, geo-referenced aerial image	• ArcGIS COTS application	World boundary coordinates	No							
	Pose Estimation	• Camera intrinsic parameters • Image correspondence points • World boundary coordinates	• Custom C++ application		No			×				
	Rectification	• Stabilized video • Transformation matrices	• Custom C++ application		No		×		×			
Other Processing Steps	Specify Zoning	• Ortho-rectified, geo-referenced aerial image	• Custom ArcGIS application		No					×		
	Specify Directionality	• Ortho-rectified, geo-referenced aerial image	• Custom ArcGIS application		No						×	
	Create Configuration File	• Stabilized raw video file names and locations • Rectified video file names • Camera intrinsic parameters • Transformation matrices • Zoning parameters • Direction polygons and points	• XMLSpy COTS application		No							×

3.1 VIDEO EDITING

Input

The inputs to this process are one or more AVI files obtained from the video collection effort. The input file settings are recommended to comply with the specifications of the data collection standards. The current specification for the input file is AVI file format at *8,000 Kb* compression bit rate in *MPEG-4* standard using the *XVID* codec.

Theory

Based on the tracking requirements, AVI files may have to be edited. Video editing may include the following:

- Splicing/truncating the video file(s);
- Appending multiple video file segments; and/or
- Decimating video frames (reducing video frame rate).

Splicing/truncating AVI files may be needed if:

- The first few frames of the data collection effort are blank due to camera limitations;
- The data are collected for much longer time than the required tracking period;
- The AVI files exceed some versions of Microsoft Windows' limitation of 2 GB size for AVI files, which may require that the source files be segmented to smaller sizes to be Windows-compliant before rectification; and/or
- The video file needs to be spliced into multiple smaller segments and be tracked by several users on separate computers.

Appending AVI files may occur if the source AVI file is spliced for rectification purposes, and then the rectified AVI files can be appended before running the NGSIM-VIDEO program on the complete data.

Decimating video frames may be needed if the data are collected in the field at a high frame rate, while the tracking will be performed at a lower frame rate.

Application

An open-source freeware program, VirtualDub, may be used for video editing.

Steps

1. Run VirtualDub;
2. For splicing/truncating AVI files:
   a. On the main screen, select *Video | Direct Stream Copy*, as shown in Figure 3.1. According to the on-line FAQ, this attribute will copy the source to destination at the same compression settings. It is a good idea to avoid recompressing data.
   b. If a file within a selected video range has to be created, select *Video | Select Range*, and input the time or frames as shown in Figure 3.2.
   c. If segments of the original file have to be created, select *File | Save Segmented AVI…*, and input the frame limit in the field at the bottom, as shown in Figure 3.3.
3. For appending AVI files:
   a. Open the first file, and append the subsequent files by selecting a new AVI file through the command *File | Append AVI Segment*.
4. For decimating AVI files:
   a. Open the file for which the frame rate has to be reduced.
   b. Select *Video | Direct Stream Copy* to create the new AVI file without recompressing.
   c. Select *Video | Frame Rate*, as shown in Figure 3.4, and check the appropriate frame rate conversion box to obtain the required output frame rate. For example, if the original AVI file has a frame rate of 30 frames/second (fps), by choosing "Process every third frame (decimate by 3)" the frame rate would become 10 fps after decimation.

Output

The output of this *Video Editing* Phase is a set of AVI files satisfying the requirements for NGSIM-VIDEO program, and having the same compression setting.

Notes

VirtualDubMod[8] is another freeware program that provides similar features as VirtualDub, and is also quite user-friendly.

[8] http://virtualdubmod.sourceforge.net.

Figure 3.1 Select Direct Stream Copy in VirtualDub

Figure 3.2 Select Range in VirtualDub

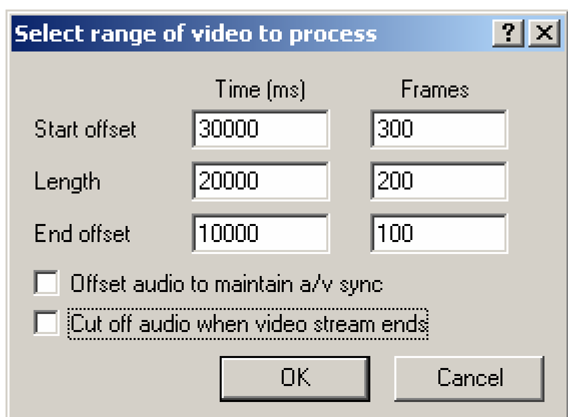


Figure 3.3 Saving Segmented AVI File in VirtualDub

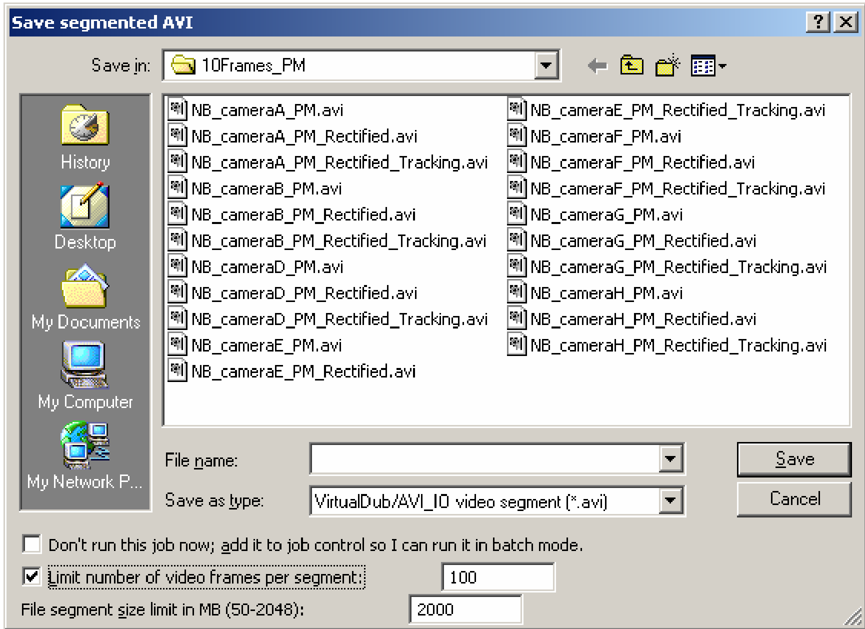
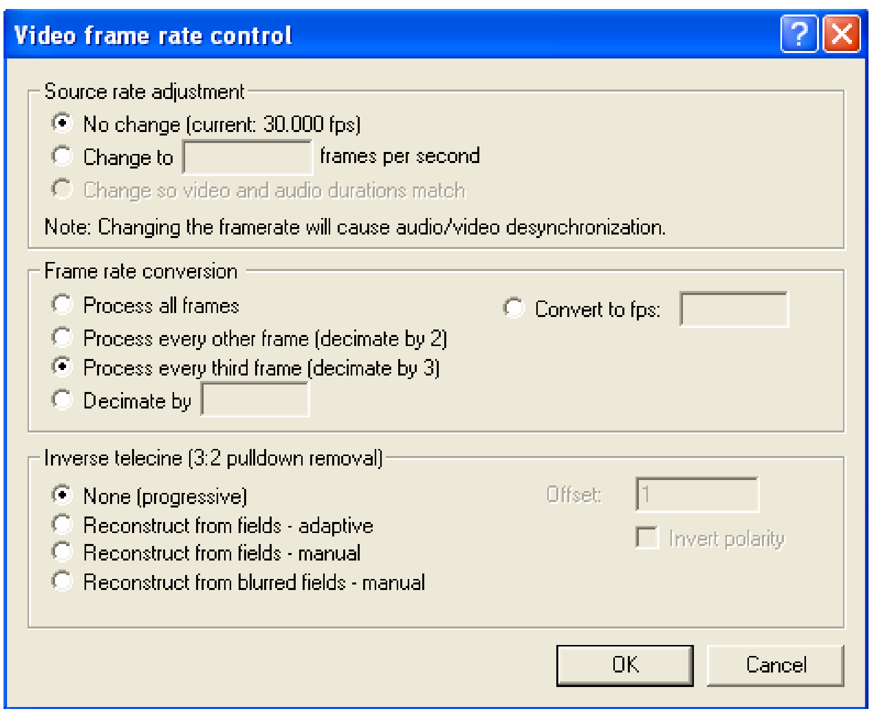


Figure 3.4 Decimate AVI File in VirtualDub



3.2 VIDEO STABILIZATION

Input

The inputs to this process are one or more AVI files obtained from the field video collection effort. The input file settings are recommended to comply with the specifications of the video requirement standards as discussed in Section 2.0.

Theory

In some cases, videos may need to be stabilized, due to unstable data collection platforms or environmental disturbances (such as wind). Video stabilization software is common and it uses relatively well understood tracking algorithms to stabilize videos for a wide variety of applications, including commercial film tracking shot applications. Several off-the-shelf software packages provide efficient video stabilization. Upper limits on the effectiveness of stabilization on extremely mobile camera shots are poorly understood, but the software that has been researched appears to work well at a few pixels/Hz of instability.

Depending on the software, the features that are tracked are automatically selected by the stabilization software or set by the user. Video stabilization software zoom-in on selected features and try to keep them at a fixed location in successive images. This can lead to cropping of the edges of useful information in a video file. Trajectory extraction requires a consistent view of the study area for the entire extraction timeframe, which means that video stabilization could reduce the amount of useful image area for trajectory extraction if the camera moved significantly. It is up to the data collector to ensure that the pertinent roadway section stays in frame over the entire time period.

For example, in Table 3.2, Frame 1 shows a selected feature in the center of the image. If the video is unstable, the output image will have its borders cropped. In the same table, the row in Frame 2 presents an example of this phenomenon, where the stabilized video crops the image.

Table 3.2 Stabilization Cropping Example

Frame	Original Video	Stabilized Video	Cropping
Frame 1: Camera centered on stabilization object (house inside the square)			None
Frame 2: Camera moved to the right, stabilization object now in upper left corner			Cropping to the top and to the right, due to camera movement

Application

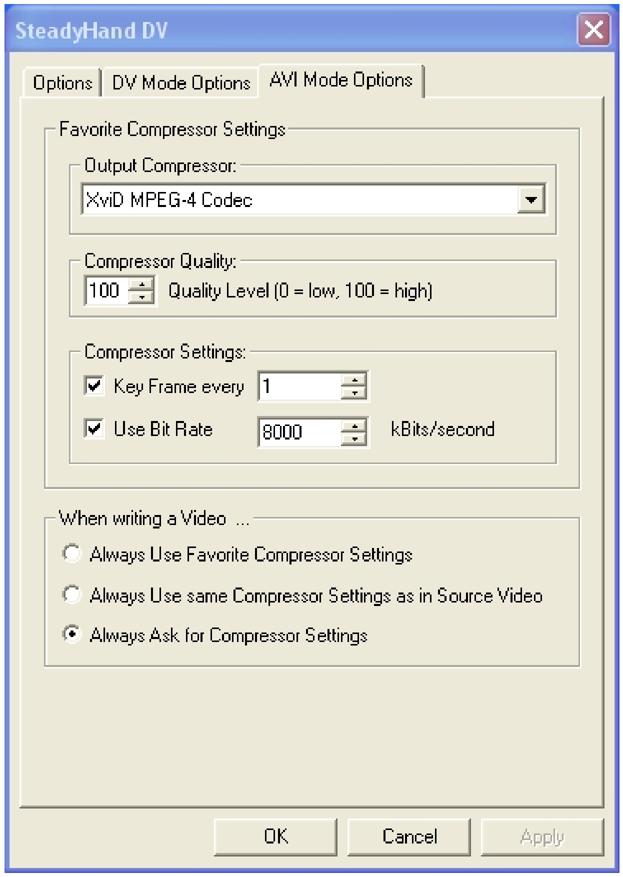
A commercial software program, SteadyHand,[9] is used for video stabilization. It is available both as a plug-in for Adobe Premiere and as a separate program. The stand-alone SteadyHand Version 2.2.0.2 was used for previous NGSIM video stabilization; however, it is not believed that any later version would pose any problem.

Steps

1. Run SteadyHand;
2. On the main screen, select the *Source Video* (*Destination Video* is set automatically);
3. On the main screen, set the *Video Edge Treatment* checkbox to *Zoom to Fill*;
4. On the main screen, set the *Motion Correction* checkbox to *Normal*;
5. Open *Settings | AVI Mode Options*, and set the compression settings as shown in Figure 3.5 (set key frame = 1 and compression rate = 8,000 Kb); and
6. Click *Start*, which will start the stabilization process.

[9] http://www.dynapel.com.

Figure 3.5 SteadyHand Compression Settings



Output

The output of the *Video Stabilization* phase is a set of stabilized AVI files that are compressed at the settings specified in video requirements in Section 2.0.

Notes

In previous tests on an Intel Xeon 3.0 GHz processor with 1 GB of RAM, SteadyHand converted video at a rate of about 9 GB per hour.

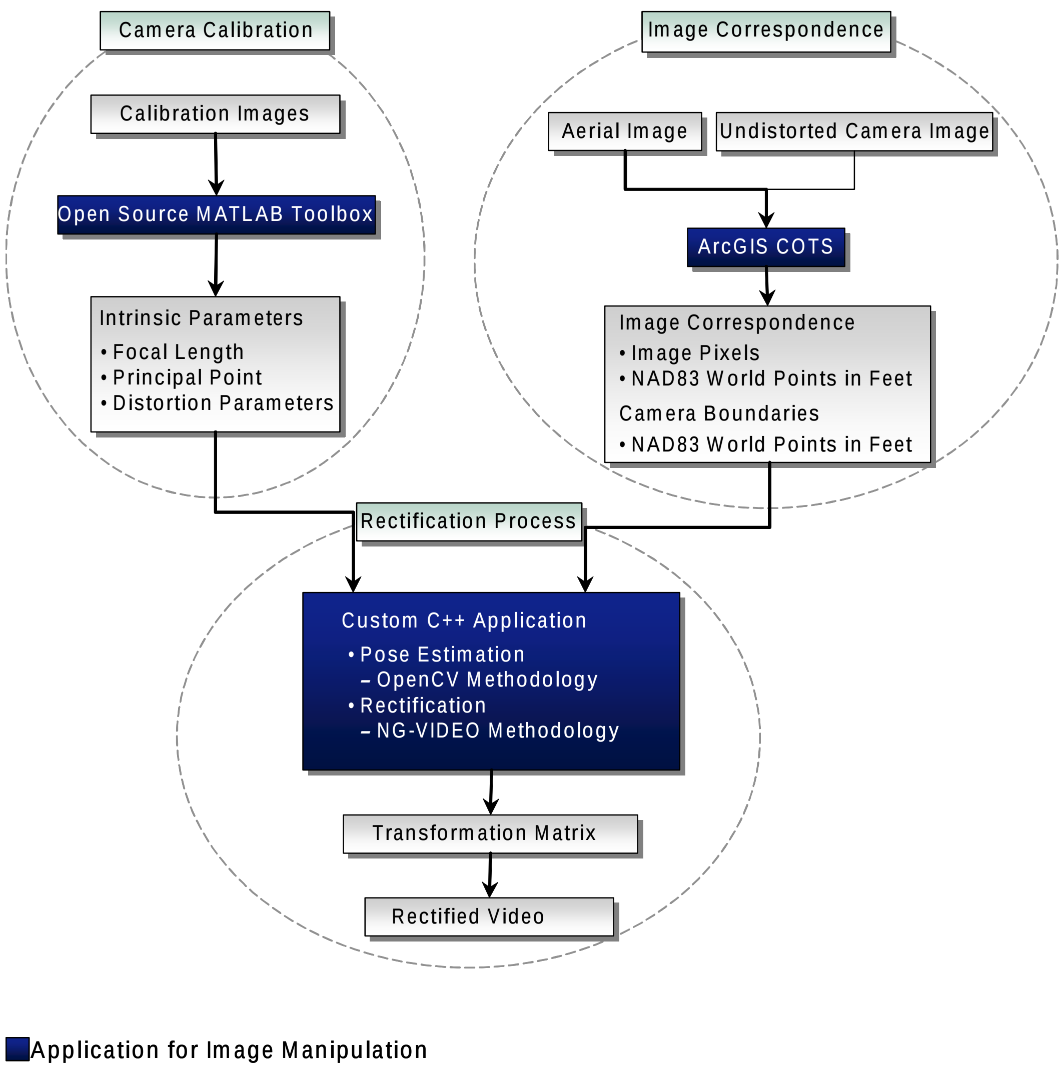
The AVI files must have recognizable features for stabilization in all the cameras. SteadyHand uses an algorithm that automatically selects the required features. Other software, such as Adobe's AfterEffects, uses alternative stabilization methodologies. These methodologies allow the user to select non-moving features for stabilizations. In our tests of unstable traffic video SteadyHand outperformed AfterEffects and, thus, we recommend SteadyHand over AfterEffects for stabilization. It is possible for some videos (i.e., video with many moving objects), that the AfterEffects' user feature selection method could outperform SteadyHand's automated feature selection method. In these cases, AfterEffects may be a better choice for video stabilization.

3.3 VIDEO RECTIFICATION

The video rectification process is illustrated in Figure 3.6. It consists of the following steps:

- Obtaining camera intrinsic parameters through camera calibration;
- Getting corresponding image and world points;
- Getting world boundary coordinates for rectification;
- Getting extrinsic parameters – the rotation and translation matrices;
- Rectifying individual camera frames using OpenCV functions; and
- Outputting the rectified frames to an output AVI file.

Figure 3.6 Video Rectification Process



Camera Calibration

Input

The inputs for this process are about 20 to 30 images of a checkerboard for each camera collected using the same camera properties (zoom, focus, etc.) as those used during the field video data collection.

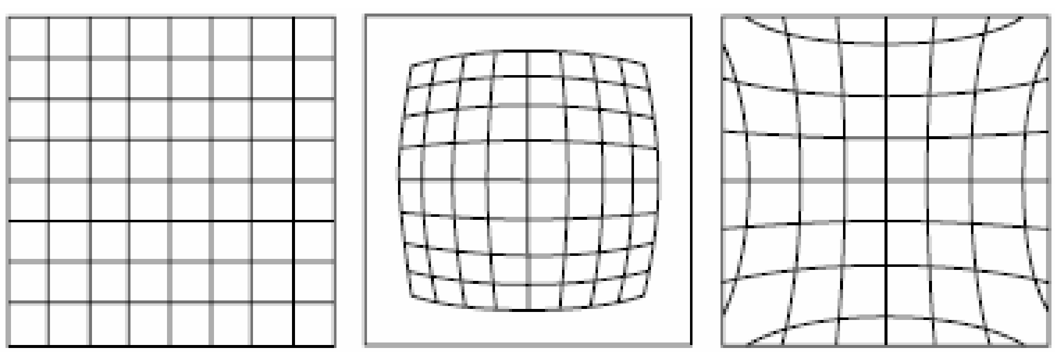
Theory

Camera calibration is the process through which the internal camera parameters which affect image process are obtained. Effective calibration allows an accurate reconstruction of the world model from the image model. The camera's intrinsic parameters that affect the reconstruction of the world image include:

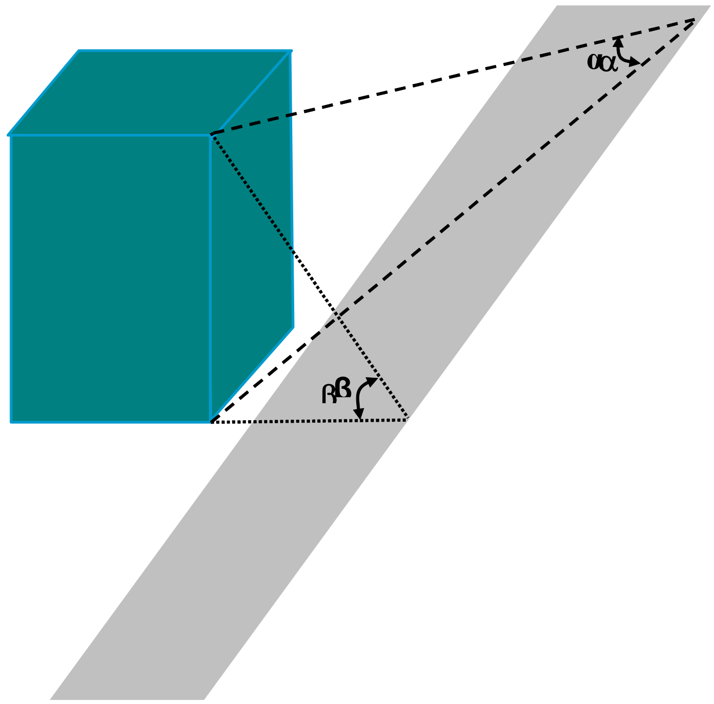
- Position of the image center – not necessarily at the center of the image;
- Focal length;
- Scaling factors for row and column pixels;
- Skew factor; and
- Lens distortion.

The effect of optical distortion in a lens is to create curved images of orthogonal objects as shown in Figure 3.7. Distortion is unique among aberrations, in that it does not blur the image. As a result, it is possible to manipulate a digital image with a mathematical formula to eliminate distortion without significantly harming the image sharpness. Typically, wide-angle lenses tend to suffer from barrel distortion and tele-lenses suffer from pincushion distortion. Both effects tend to be stronger at the extreme ends of zoom lenses.

Figure 3.7 The Original Grid, Effect of Barrel (Center Image), and Effect of Pincushion (Right Image)



There are three popular camera calibration algorithms: Tsai and Heikkilä's world-reference-based approaches and Zhang's planar calibration method. Of the three, Zhang's method takes advantage of planar constraints of the calibration pattern and requires only relative measurements between adjacent calibration points, which can be accomplished at very high accuracy with trivial effort. Hence, Zhang's methodology is used for the calibration process, while the output camera intrinsic parameters are based on a combination of Zhang's and Heikkilä's parameters.



Application

The calibration is applied through an open source MATLAB toolbox developed by Jean-Yves Bouguet.[10] The main initialization phase, including the initial estimation of the planar homographies in the toolbox, is based on Zhang's methodology, while the intrinsic model is based on Heikkilä and Silven's camera model, which includes two extra distortion coefficients corresponding to tangential distortion. The MATLAB toolbox calibration procedure provides the intrinsic parameters output similar as illustrated in Figure 3.8.

[10]http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

Figure 3.8 Intrinsic Camera Parameters from MATLAB Toolbox

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 657.30254   657.74391 ] ± [ 0.28487   0.28937 ]
Principal point:       cc = [ 302.71656   242.33386 ] ± [ 0.59115   0.55710 ]
Skew:             alpha_c = [ 0.00042 ] ± [ 0.00019  ]   => angle of pixel axes = 89.97595 ± 0.01092 degrees
Distortion:            kc = [ -0.25349   0.11868   -0.00028   0.00005  0.00000 ] ± [ 0.00231   0.00942   0.00012   0.00012  0.00000 ]
Pixel error:          err = [ 0.11743   0.11585 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

Steps

1. Unibrain's Fire-i[11] software provides control of multiple firewire cameras. This software is capable of grabbing a set number of frames at preset time limits (Figure 3.9). This feature of the software is used to obtain camera images. A checkerboard pasted on a cardboard is moved in front of the camera and about 20 to 30 images are obtained at the set camera properties which match the camera properties applied for the field video collection.

2. The process of camera calibration is detailed in one of the calibration examples on Bouguet's web site.[12] This procedure must be applied to get the camera's intrinsic parameters. It is strongly recommended that users go through the process before making real calibrations. The calibration technique requires the camera to observe a planar pattern, a checkerboard as shown in Figure 3.10, shown at a few (recommended at least 20) different orientations. Either the camera or the planar pattern can be freely moved; however, setting camera firmly while moving the planar pattern is easier to operate and, therefore, is recommended. Both tangential and radial lens distortions are modeled by the toolbox. The intrinsic camera parameters will be provided as input for the video rectification process.

---

[11] http://www.unibrain.com.

[12] http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html.

Figure 3.9 Frame Capture Setting in Fire-i

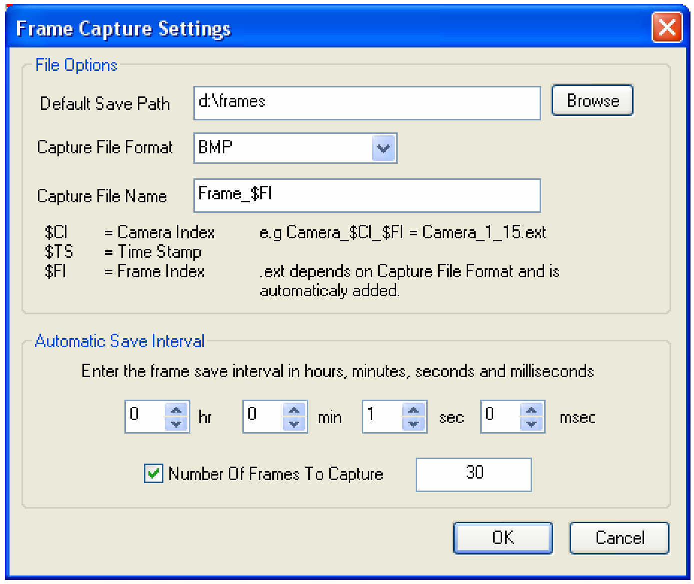
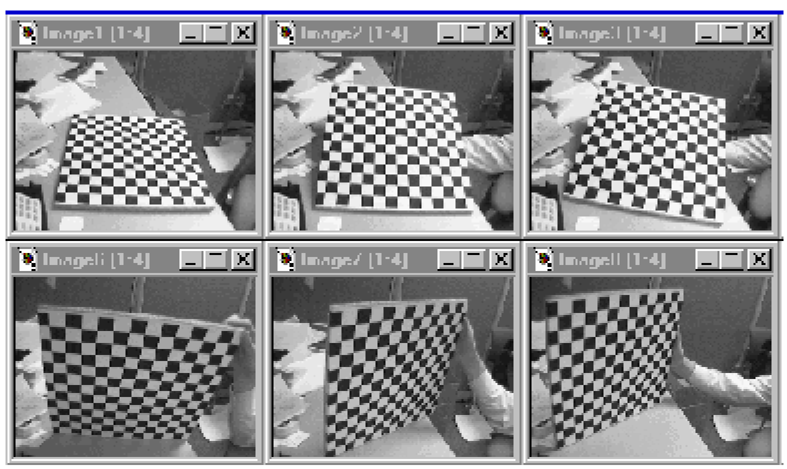


Figure 3.10 Checkerboard Images



Output

The output of this process is the calib_results.m file, which provides the intrinsic parameters for each camera.

Notes

- The checkerboard pattern must be glued flatly on a rigid surface;
- Twenty to 30 images must be used for the calibration; and
- Camera settings must be the same as the ones used during field video data collection.

Image Correspondence

Input

The inputs required for image correspondence are an ortho-rectified aerial image of the data collection location and an undistorted camera image from the data collection video. The ortho-rectified image is geo-referenced (tiff + tfw) to provide NAD83 coordinates (in feet).

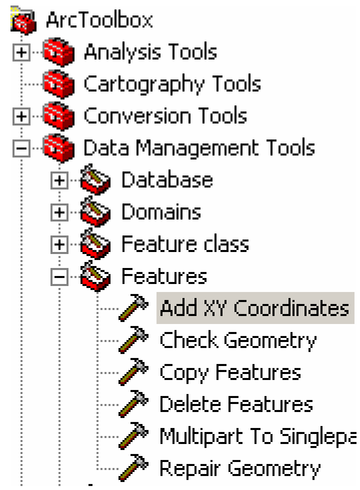
Theory

Image correspondence is the process where identifiable features in the world aerial image and the undistorted camera image are mapped. This correspondence matching is necessary for obtaining rotation and translation matrices. This step provides the means for finding the camera's extrinsic or external parameters (position and orientation of the image relative to a world coordinate system). The selection of features is based on manually matching lane and pavement markings, fixed structures, etc., between the camera image and ortho-rectified aerial image.

Application

The ArcGIS suite of software is used as the base platform for obtaining the feature dataset for image correspondence. Specifically, ArcMap is used for opening the ortho-rectified aerial image and selecting point features. The X and Y coordinates of the selected feature can be obtained by using ArcToolbox in ArcMap. To do so, open ArcToolbox through Windows menu in ArcMap. In ArcToolbox, select Data Management Tools, expand Features, and then click Add XY Coordinates, as illustrated in Figure 3.11.

Figure 3.11 Extract X and Y Coordinates Using ArcToolbox



Steps

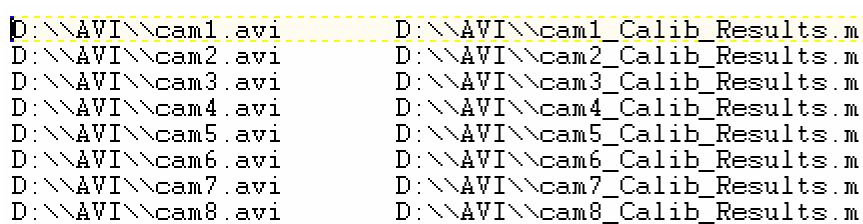
The image correspondence step provides the matching features between an ortho-rectified aerial image and an undistorted camera image required for getting camera's extrinsic parameters. The process is described below.

- An undistorted camera image can be obtained by running undistort_NGVideo.exe, a custom program developed by the NGSIM team. The program can be run from the command line:

  >> undistort_NGVideo <aviList_Undistort.txt>

  The aviList_Undistort.txt file, shown in Figure 3.12, lists the AVI files and the corresponding Calib_Results.m files separated by *tab* character. The Calib_Results.m file is the output calibration results file from the MATLAB Toolbox. The program will create bitmap files by undistorting the third frame in each AVI file. The bitmap for each AVI file is created in the same directory and with the same filename (bmp extension).

Figure 3.12 Text File Input for undistort_NGVideo Program



```
D:\\AVI\\cam1.avi	D:\\AVI\\cam1_Calib_Results.m
D:\\AVI\\cam2.avi	D:\\AVI\\cam2_Calib_Results.m
D:\\AVI\\cam3.avi	D:\\AVI\\cam3_Calib_Results.m
D:\\AVI\\cam4.avi	D:\\AVI\\cam4_Calib_Results.m
D:\\AVI\\cam5.avi	D:\\AVI\\cam5_Calib_Results.m
D:\\AVI\\cam6.avi	D:\\AVI\\cam6_Calib_Results.m
D:\\AVI\\cam7.avi	D:\\AVI\\cam7_Calib_Results.m
D:\\AVI\\cam8.avi	D:\\AVI\\cam8_Calib_Results.m
```

- Both the aerial image and the undistorted camera image are opened in ArcGIS.
- Two shapefiles (feature type: points), one for the world points on the aerial image and the other for image points on the distorted image, are created and opened in ArcGIS.
- A point is added in the image point's shapefile on a recognizable feature in the undistorted camera image. A corresponding point is added in the world points shape file at the same feature in world image. This process is repeated for multiple points. It is recommended that a minimum of eight corresponding points are selected. The higher the number of corresponding points and the wider the spatial distribution, the better the results of the rectification process. An example of the feature correspondence is shown in Figure 3.13. The world points 0 to 3 are world boundaries of the region of interest (ROI); world points 10 to 21 match with image points 0 to 11.
- ArcGIS provides the feature to save these points to a text file (open attribute table | options | export). Both the world point table and the image point table can be saved to text files.
- A single image correspondence file containing X_w, Y_w, Z_w (3D world points) and u, v (2D image points) is created from the two files.

Figure 3.13 Aerial Image with World Points and Undistorted Image with Image Points

Output

The output of this process is the image correspondence file shown in Figure 3.14.

Figure 3.14 Image Correspondence File

worldX	worldY	worldZ	Image_X	Image_Y
6451590.98	1872823.19	0	209.84	112.26
6451669.34	1872799.13	0	19.00	188.79
6451597.26	1872862.46	0	279.73	202.33
6451507.47	1872940.98	0	551.74	217.82
6451686.72	1872827.86	0	57.10	281.29
6451644.05	1872850.42	0	182.47	253.18
6451614.42	1872890.83	0	317.41	287.30
6451570.56	1872914.61	0	425.82	263.29
6451541.32	1872955.18	0	542.53	294.38

Notes

The image pixel coordinates must have the origin at the top left corner of the image with *x* increasing from left to right, and *y* increasing from top to bottom. If the coordinate system does not correspond to this, translation needs to be applied to make the top left corner of the image as the origin.

World Boundary

Input

The input required for world boundary is the ortho-rectified aerial image, which is geo-referenced (tiff + tfw) to provide NAD83 coordinates (in feet).

Theory

World boundary establishes the boundaries of the rectification in world coordinates. These world boundaries allow the establishing of the image region of interest (ROI). While the image ROI can be set in the undistorted camera image directly, it gets quite difficult to set the boundaries across various camera images. Setting world boundaries for all the cameras in a single aerial image (or a set of aerial images that are mosaicked using the NAD 83 coordinate system) is relatively trivial.

Application

The ArcGIS suite of software is used as the base platform to obtain the feature dataset for image correspondence. Specifically, ArcMap is used for opening the ortho-rectified aerial image and selecting polygon features.

Steps

World boundary polygons are created separately for each rectified AVI file. The world boundary setting for the rectified AVI file requires the user to select the image ROI in each camera, and draw a polygon on the aerial image to represent this area. This process is repeated for all cameras. A single delineator must be used between camera boundaries; this will result in smooth mosaicking of the camera images.

A shapefile of polygon type is created in ArcCatalog and opened in ArcMap. The boundary polygons can be obtained by drawing polygons representing the camera boundaries in the shapefile on top of the aerial image in ArcGIS software. Using the Visual Basic macro code listed in Appendix A, the vertices of the polygon can be extracted from ArcGIS software.

Output

The output of this process is a world boundary points file as illustrated in Figure 3.15. The first set of coordinates is the world boundary coordinates to be set for the rectified images. The second set of coordinates (separated by a line break) was used for the rectified+tracking images, which include overlaps between cameras. However, the NGSIM team found it is not necessary to use rectified+tracking images for the NGSIM-VIDEO processing.

Figure 3.15 World Boundary File

```
worldX          worldY          worldZ
6451519.21      1872993.35      0
6451698.59      1872837.62      0
6451644.74      1872775.63      0
6451467.22      1872929.51      0

6451519.21      1872993.35      0
6451698.59      1872837.62      0
6451644.74      1872775.63      0
6451467.22      1872929.51      0
```

Notes

- A single boundary delineation must be used between cameras to allow a seamless mosaic between the video streams of rectified cameras. The rectified image widths must be a multiple of 8, which is set internally by the rectification program. The rectified images also need to have an even number of widths and heights. These again are set internally in the rectification program. The user is not required to set the world boundary widths and heights to even numbers or multiples of 8.
- Use the Visual Basic macro code listed in Appendix A to extract X and Y coordinates when the shapefile type is polygon or polyline. However, if the shapefile is in point style, users can also acquire both X and Y coordinates using ArcToolbox in ArcMap, as illustrated in Figure 3.11.

Pose Estimation

Input

The inputs for this step are: 1) internal camera calibration parameters, and 2) image correspondence points.

Theory

The image rectification step can be divided into the pose estimation and rectification. In this application, the camera has already been calibrated using the MATLAB toolbox; hence, a model of the internal camera parameters is available. This step, called pose estimation, needs to recover the six parameters relative to the position and orientation of the camera (three rotation angles and three translation values).

The *extrinsic* camera parameters describe spatial relationship between the camera and the world; they are:

- Rotation matrix; and
- Translation vector.

They specify the transformation between the camera and world reference frames.

For a pinhole camera, the relationship between a *3D* point *M* and its image projection *m* is given by the formula

$$m = C\,[Rt]\,M$$

Where, C is the camera's intrinsic matrix:

$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where, (c_x, c_y) are coordinates of the principal point; and

(f_x, f_y) are focal lengths for x and y axes.

(R, t) are extrinsic parameters: the rotation matrix *R* and the translation vector *t* that relate the world coordinate system to the camera coordinate system:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \; t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

The theory behind m = C [Rt] M is described below.

Figure 3.16 shows a camera with center of projection *C* and the principal axis parallel to *Z* axis. Image plane is at focus and, hence, focal length *f* away from *C*. A 3D point (x_s, y_s, z_s) is imaged on the camera's image plane at coordinate (x_i, y_i). First, find the camera calibration matrix C, which maps the 3D world point to a 2D image point. Using similar triangles:

$$\frac{f}{z_s} = \frac{x_i}{x_s} = \frac{y_i}{y_s}$$

Which gives:

$$x_i = \frac{f\ x_s}{z_s} \text{ and } y_i = \frac{f\ y_s}{z_s}$$

Figure 3.16 Camera Central Projection

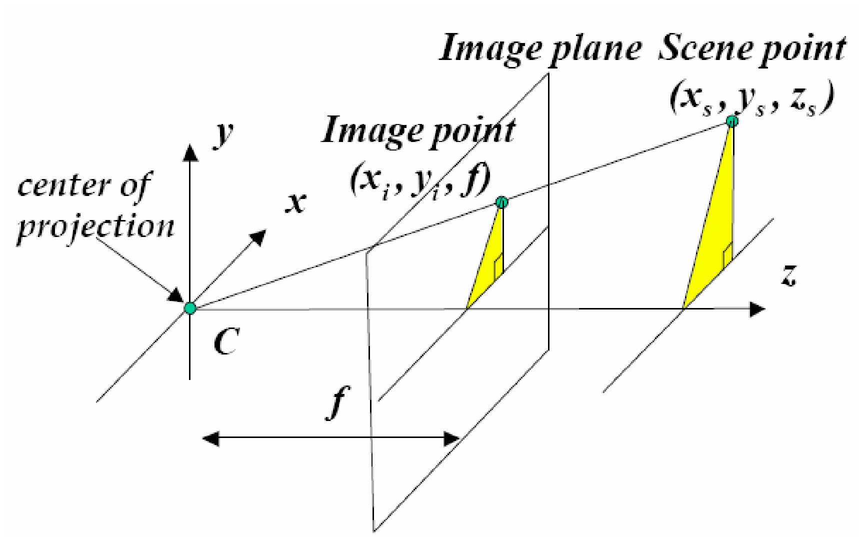


Figure 3.17 provides the pixel coordinate transformation. Suppose the origin of the 2D image does not coincide with where the Z axis intersects the image plane. We now need to translate the image point to the desired origin. Conducting the translation, we get:

$$x_{pix} = k_x x_i + c_x = f\ k_x \frac{x_s + z_s c_x}{z_s}$$

$$\text{and } y_{pix} = k_x y_i + c_y = f\ k_y \frac{y_s + z_s c_y}{z_s}$$

where, (c_x, c_y) define the image center and (k_x, k_y) are scaling factors.

Using homogeneous coordinates for 2D image plane coordinates, we can write the equation as:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

Figure 3.17 Transformation from Length to Pixels

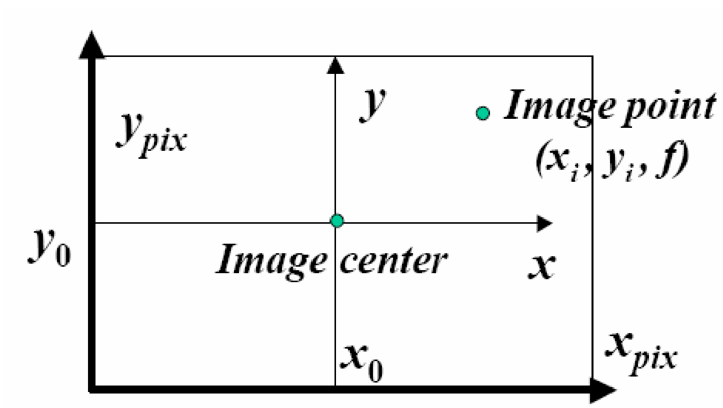
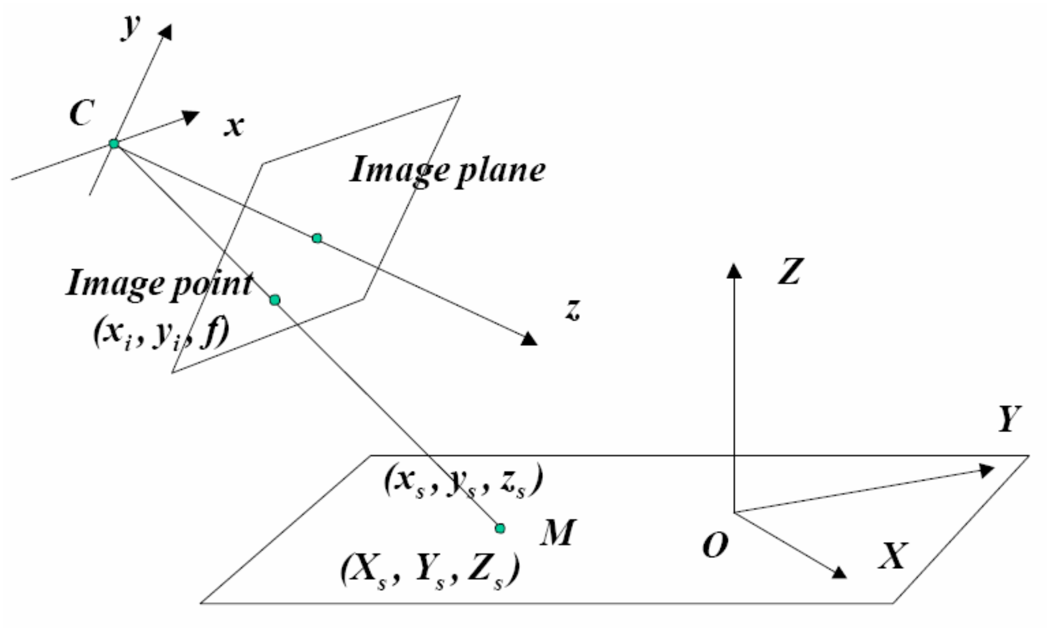


Figure 3.18 illustrates the condition where the camera does not have its center of projection at (0,0,0) and is oriented in an arbitrary fashion (not necessarily *z* perpendicular to the image plane); in this case, rotation and translation need to be applied to the camera coordinate systems to coincide with the configuration in Figure 3.14. Let the camera translation to the origin be given by $t = [t_1\ t_2\ t_3]$ and rotation by:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Figure 3.18 Camera Coordinates to World Coordinates



Then, the relationship between image coordinates and world coordinates is given by:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix}$$

Combining camera projection and coordinate transformation matrices into a single matrix, we get:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} + \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} \right)$$

or m = C [Rt] M

Application

A custom program using OpenCV library was developed in C++ for obtaining the rotation and translation matrices and conducting the rectification (described in the next section). OpenCV's cvFindExtrinsicCameraParams_64d and cvRodrigues functions are used for obtaining the rotation and translation matrices.

Steps

The pose estimation step provides the rotation and translation matrices necessary for converting between image and world coordinates. The custom C++ software, using OpenCV 4.0 functions, first calculates the rotation and translation matrices, and then conducts the rectification. The steps for running this program are provided in the rectification section below.

Output

The pose estimation step occurs within the image rectification program, while the output from this step is the rotation and translation matrices that are used internally by the rectification program.

Notes

- Pose estimation works only if the camera intrinsic parameters are known beforehand.

Rectification

Input

The last step in the *Video Rectification* process is the rectification of the video. The inputs for the rectification are: the stabilized AVI file, camera's intrinsic parameters file, the image correspondence points, the camera world boundary points, and the rotation and translation matrices.

Theory

Real-world distances are not represented uniformly on an image. For example, an inch measured in one part of the photograph could relate to a much longer distance than an inch measured in another part of the photograph. Rectification is the process used to remove these sources of distortion to equilibrate photo units with real-life distances. Once an image has been rectified, the distances will be representative of world distances. It should be noted that, while distances in the image plane are rectified, the vertical distances are not corrected. For example, the sides of a building or truck would still contain distortion.

Rectification consists of setting an image ROI from world boundary points, and rectifying this region to form a rectified image. This is achieved by interpolating and copying each image pixel in the ROI from the source image to the rectified image.

Application

A custom application, rectify_NGVideo.exe, was developed by the NGSIM team for the rectification process. The inputs for this program are the stabilized AVI file, the camera intrinsic parameters file (calib_results.m), the world to image correspondence file, and the world boundary points file. The output of the program is a rectified video file.

Steps

Please note that the Microsoft Windows AVI size limit of 2 GB requires that input files for the rectification process have to be broken down into smaller files before conducting pose estimation and rectification. The size of the input files must be set that the output rectified AVI files are less than 2 GB.

The steps involved in rectifying an image consist of running the custom C++ program, rectify_ NGVideo.exe, at command line, as follows:

1. >> rectify_NGVideo <aviList_Rectify.txt>
2. The aviList_Rectify.txt file, as illustrated in Figure 3.19, lists in each line: the AVI file, the corresponding Calib_Results.m file, the image correspondence file, the world boundary file, and the pixels per foot of the output image. The inputs must be separated by a *tab* character. The program will create rectified files by cropping the images to the set world boundary and keystoning the images in the AVI file. The program creates a rectified AVI file with "_rectified" added to the input AVI filename, and a rectified+tracking AVI file, which is the same as the rectified AVI file, with "_rectified_tracking" added to the filename. The files are created in the same location as the original AVI files.

Figure 3.19 Text File Input for rectify_ NGVideo Program

```
d:\NB_CamF.avi   d:\Calib_Results_CamF.m d:\Correspondence_AM_CamF.txt   d:\Boundary_CamF.txt    2.0
d:\NB_CamA.avi   d:\Calib_Results_CamA.m d:\Correspondence_AM_CamA.txt   d:\Boundary_CamA.txt    2.0
d:\NB_CamB.avi   d:\Calib_Results_CamB.m d:\Correspondence_AM_CamB.txt   d:\Boundary_CamB.txt    2.0
d:\NB_CamC.avi   d:\Calib_Results_CamC.m d:\Correspondence_AM_CamC.txt   d:\Boundary_CamC.txt    2.0
d:\NB_CamD.avi   d:\Calib_Results_CamD.m d:\Correspondence_AM_CamD.txt   d:\Boundary_CamD.txt    2.0
d:\NB_CamE.avi   d:\Calib_Results_CamE.m d:\Correspondence_AM_CamE.txt   d:\Boundary_CamE.txt    2.0
```

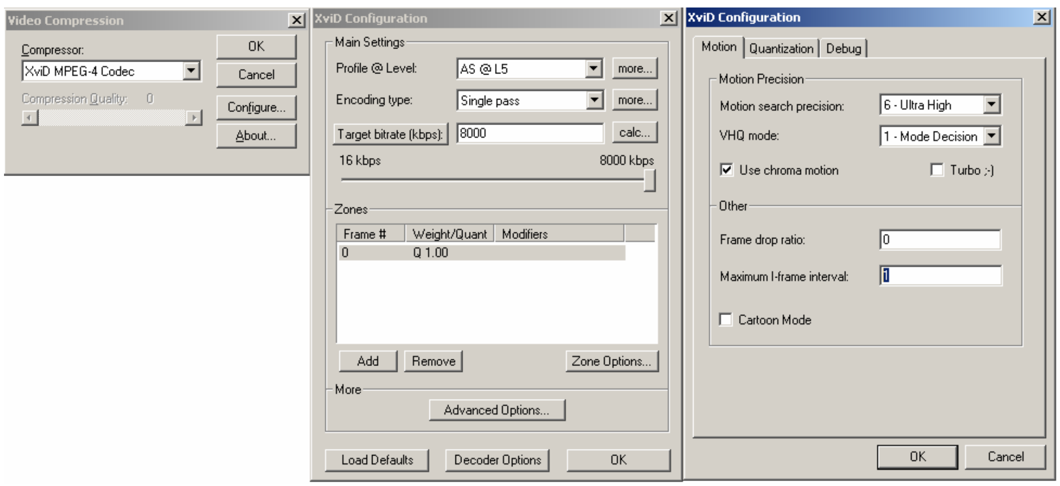
The process that rectify_ NGVideo.exe follows is listed below.

1. Open the stabilized source AVI file.
2. Grab each frame, and
   a. Undistort the image frame using OpenCV's cvUndistortInit and cvUndistort functions.
   b. Get the rotation and translation matrices from the pose estimation step in the rectify.exe program.
   c. Get the image ROI corresponding to the world boundary points by applying the transformation matrices.
   d. Create destination (rectified, and rectified+tracking) image frames by interpolating pixels from the undistorted source image within the region of interest.

e. Add the rectified and rectified+tracking images to two different AVI writers using OpenCV 4.0. The codec for the AVI writer is user-defined. The user has to set the codec and its configuration in the dialog box that pops-up when the program is running. Use the same configurations set in the input (stabilized) AVI file. Please note that the *Maximum I-frame interval in Advanced XVID options must be set to 1* from the default 300. The three dialog boxes for setting the necessary XVID settings are illustrated in Figure 3.20.

f. Repeat the process for the all the frames in the input AVI file.

Figure 3.20 XVID Codec Configuration Settings



Output

The outputs of this step are rectified video files. In addition, a text file with transformation matrices is also created. The text file, *transformMatrices.txt* is created in the same location as the input text file for rectification (Figure 3.19). The file contains three sets of transformation matrices: 1) the first set of transformation matrices converts between the undistorted source image and world coordinates, 2) the second set converts between rectified image file and world coordinates, and 3) the third set converts between rectified+tracking image file and world coordinates. Figure 3.21 shows a snapshot of the source AVI with the ROI. As illustrated in Figure 3.22, the ROI in the source AVI is extracted and interpolated to provide a rectified AVI file.

Figure 3.21 Source AVI Showing Image ROI

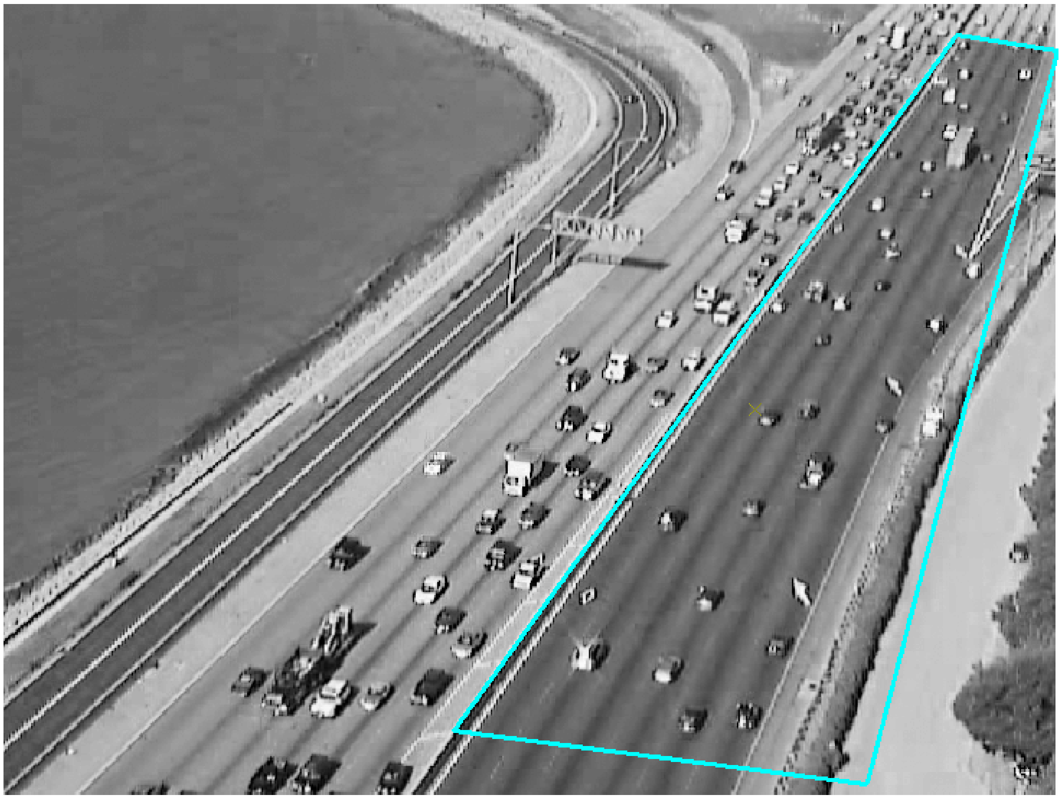


Figure 3.22 Rectified AVI of the ROI

Notes

- Whenever XVID codec is used, the user should manually set the Maximum I-frame interval in Advanced XVID options to 1 from the default 300.
- The rectification process (rotation and translation) automatically takes into consideration different zoom settings in the camera used for data collection based on the camera calibration file.

- In the rectification process, we can set the rectify_NGVideo.exe program to read alternate frame, etc., based on the NGSIM-VIDEO requirements. Currently, this option is hard-coded in the program, but can be made user-defined. Dropping frames can also be achieved through VirtualDub, an open-source, freeware program for manipulating video files. In VirtualDub, use Video | Frame Rate | Change to *X* Frames option to drop frames.
- The rectification process transforms only the world 2D coordinates to image 2D coordinates (i.e., the process does not transform z-value of the ground). Transformations to get the correct world 3D coordinates have to be applied in the post-processing phase, if necessary.

3.4 CONFIGURATION FILE

This section describes processes that determine the configuration settings for NGSIM-VIDEO. The parameters provided in the NGSIM-VIDEO settings allow the detection and tracking processes in NGSIM-VIDEO to work efficiently.

Zoning

Input

An aerial image of the location is the input for this processing step.

Theory

NGSIM-VIDEO requires the specification of the **entry (detection) and exit (drop)** zones of the study area; **camera overlap** zones, where camera stitching needs to occur; and **occlusion** zones, where vehicles are occluded and later on have to be re-identified (such as being blocked by trees). Setting these zones accurately can greatly enhance the performance of NGSIM-VIDEO.

Application

The ArcGIS suite of software is used as the base platform for setting the boundary polygons for each of these zones. Specifically, ArcMap is used for opening the ortho-rectified aerial image and drawing the polygon features.

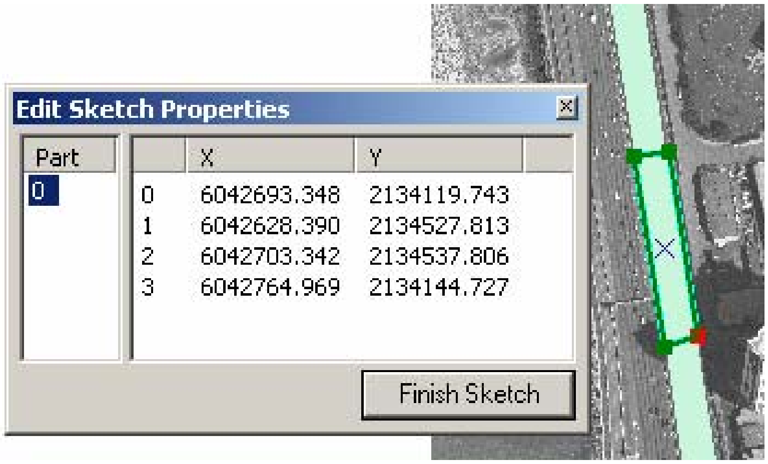
Steps

The boundary polygons for each zone can be obtained by drawing polygons on top of the aerial image in the ArcGIS software. Using the Visual Basic macro code listed in Appendix A, the vertices of the polygon (as shown in Figure 3.23) can be extracted from the ArcGIS software. The values will then be manually copied from the output text file that the ArcGIS macro provides to the configuration file.

Output

The output of this step is a text file containing the polygon boundaries of each zone.

Figure 3.23 Polygon Vertices for Zoning



Notes

- Camera overlap zones in the aerial image must be set in accordance with the actual overlap occurring in the cameras during data collection.

Direction Polygons and Points

Input

An aerial image of the location is the input for this processing step.

Theory

NGSIM-VIDEO, the detection and tracking software, assumes a constant direction of travel. This condition will not be true at intersections, where the direction of travel changes significantly. To account for such scenarios, directional polygons as well as multiple points within the directional polygon indicating the moving directions need to be provided.

Application

The ArcGIS suite of software is used as the base platform for obtaining the direction polygons and directional points for direction vectors. Specifically, ArcMap is used for opening the ortho-rectified aerial image and drawing polygons and points.

Steps

The direction polygons and directional points can be obtained by drawing polygons and points, respectively, on top of the aerial image using the ArcGIS software. Using the Visual Basic macro code listed in Appendix A, the vertices of the polygon and points can be extracted from the ArcGIS software. The points indicating the direction of travel must be drawn sequentially in the direction of travel. The driving direction polygons are useful for tracking. The values will be manually copied from the output text file that the ArcGIS macro provides to the configuration file.

Output

The output of this step is a text file containing the direction polygon boundaries and directional points for each directional polygon, giving the direction of travel.

Notes

A potential issue is the presence of overlapping polygons. Overlapping direction polygons with different directional points are provided as inputs to NGSIM-VIDEO. NGSIM-VIDEO needs to correctly interpret the driving direction for each vehicle in the presence of overlapping directional polygons. If not, the user is able to change to the correct direction polygon during tracking, as discussed in Section 4.0.

Shadow Parameters

Input

The time and date of data collection and location of the study area in latitude and longitude are the inputs for this step.

Theory

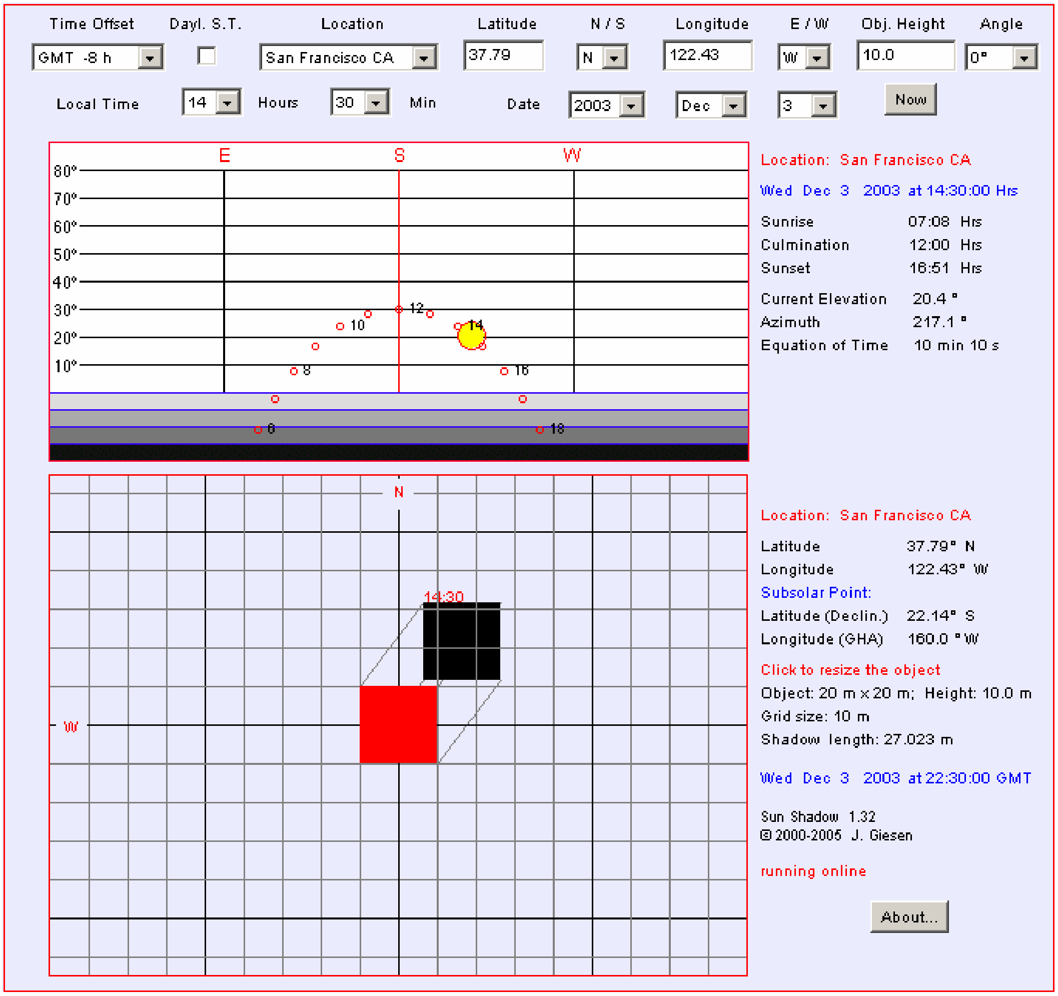
The shadow of vehicles can affect the detection algorithm in NGSIM-VIDEO. Hence, the NGSIM-VIDEO program takes shadow parameters as inputs and conducts internal manipulation to provide improved detection and tracking.

Application

The shadow parameters are obtained from a java applet[13] available online as shown in Figure 3.24. The applet takes the latitude, longitude, date, and time of the data collection and provides the azimuth angle and current elevation as the output.

[13] http://www.jgiesen.de/sunshadow/.

Figure 3.24 Shadow Applet



Steps

- Open the java applet and provide the necessary inputs (location or latitude and longitude,[14] and time); and
- Use the azimuth (a) and current elevation (e) provided by the applet to obtain the input 3D vector for the NGSIM-VIDEO:

  If x = sin(a), y = cos(a), and z = sin(e)

  The 3D vector provided as an input to NGSIM-VIDEO is:

  $(-cx, -cy, -z)$,

  where, $c = \sqrt{1 - z^2}$

Output

The output of this step is a 3D vector that will be provided as an input to the NGSIM-VIDEO.

Notes

The North in the world coordinate system used by the data collection effort must match the true North in the shadow applet (based on longitude and latitude). The shadow parameters must be used for the NGSIM-VIDEO processing if the vehicles themselves create shadows on the roadway. Do not use shadow parameters if the adjacent structures are the ones creating the shadows.

Frame Offset

Input

The inputs needed to obtain the frame offset are the camera AVIs and the camera overlap region.

Theory

NGSIM-VIDEO tracks vehicles in each camera separately. The stitching of vehicle positions across cameras occurs, either by a hand-off between cameras during tracking or by post processing the vehicle positions. If the camera frames are not synchronized, the vehicle will not be present in the same location in the same frame ID in the two camera views. For example, if an offset of 1/15th second is present between two cameras during data collection, a vehicle in Camera 1 will not be positioned in the same place until about two frames later in Camera 2. In situations where synchronization is not accurate, the user must specify the frame offset for NGSIM-VIDEO to accurately stitch the vehicles. The

[14] www.maporama.com provides latitudes and longitudes for a given address.

Frame Offset setting specifies the camera synchronization errors in data collection to NGSIM-VIDEO, so that the necessary corrections can be applied while performing stitching of vehicle positions across cameras.

Application

The frame offset can be gathered by extracting trajectories for each camera separately through NGSIM-VIDEO (in world coordinates) and running a linear programming model with the objective function minimizing the difference in vehicle positions across the two cameras. The frame offset, at which the vehicle positions in the overlap region have minimal difference, is the frame offset to be provided as an input for stitching vehicle trajectories across cameras. This minimization can be done using Excel or MATLAB software. The frame offset can also be obtained by drawing a time-space diagram of vehicle trajectories and matching the trajectories.

Steps

If automatic synchronization hardware is used, as stated in Table 2.1, the offset between cameras for field data collection would be zero. In the unlikely scenario that the camera synchronization requirement is not met, the frame offset can be measured in two ways:

- The vehicle positions of multiple vehicles in the overlap area of the two cameras are minimized through a simple linear program. The frame offset is the value at which the difference in vehicle positions is minimal. This calculation can be conducted using Microsoft Excel or MATLAB.
- A time-space diagram of vehicle trajectories (time on x-axis and distance on y-axis) of multiple vehicles in the camera overlap region is drawn for each camera separately. The trajectory diagrams are manually moved next to each other in time, and the time offset where the majority of the vehicle trajectories across the two cameras match will be provided as the frame offset to the NGSIM-VIDEO.

Output

The output of this step is a frame offset value for each pair of cameras.

Notes

A frame offset of zero is ideal for better NGSIM-VIDEO performance. Hence, the camera synchronization requirement must be applied stringently during the data collection.

Configuration File

Input

The inputs for the configuration file are the schema file for process input and the necessary data from the data collection effort to run the NGSIM-VIDEO program.

Theory

The configuration file, in xml format, is created to serve as a documentation standard for recognizing all the necessary inputs for the NGSIM-VIDEO program.

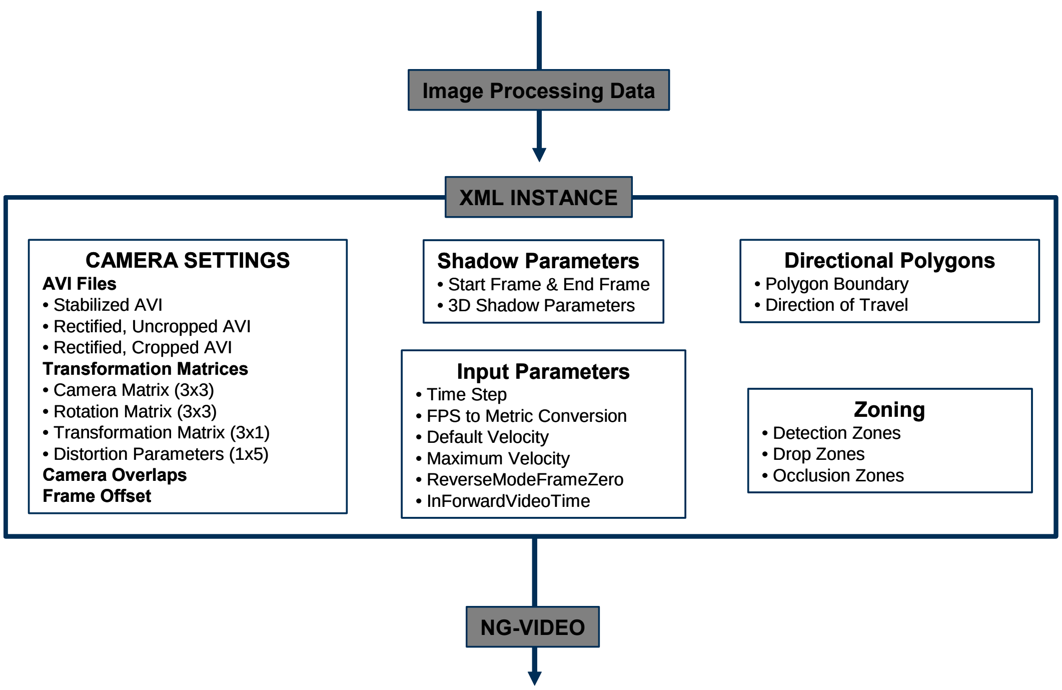
Application

XMLSpy is used to create the schema (xsd) file. No tool is created for populating the xml file. The configuration file is created manually from the input data and the data obtained from the processing tools.

Steps

The input configuration file in xml instance for NGSIM-VIDEO program is described in Figure 3.25.

Figure 3.25 XML Instance Data



The data are obtained as follows:

1. AVI files
   i. The original AVI files are the stabilized files obtained from the stabilization processing step;
   ii. The cropped rectified AVI files are obtained from the rectification processing step; and
   iii. The cropped rectified+tracking AVI files are obtained from the rectification processing step.
2. Transformation matrices
   i. The camera's intrinsic and extrinsic matrices are obtained from the calibration and pose estimation steps, respectively.
3. Input parameters
   i. The time step is obtained from the frame rate of the AVI file given as input to the NGSIM-VIDEO program.
4. Shadow parameters
   i. A 3d vector is obtained from the shadow parameters estimation step. If the angle of the sun varies across the time of study, different time periods and corresponding 3D shadow vectors need to be specified.
5. Zoning
   i. Detection zones, drop zones, and occlusion zones are specified in world coordinates.
6. Direction polygons and points
   i. The polygon boundary and directional points are specified using ArcGIS software and an ortho-rectified aerial image.

Output

The output of this step is a valid and consistent xml file containing all the specified configuration data for running NGSIM-VIDEO.

Notes

The configuration file must be checked for consistency with the schema file and be validated before providing it as an input to the NGSIM-VIDEO program.

4.0 NGSIM-VIDEO Processing

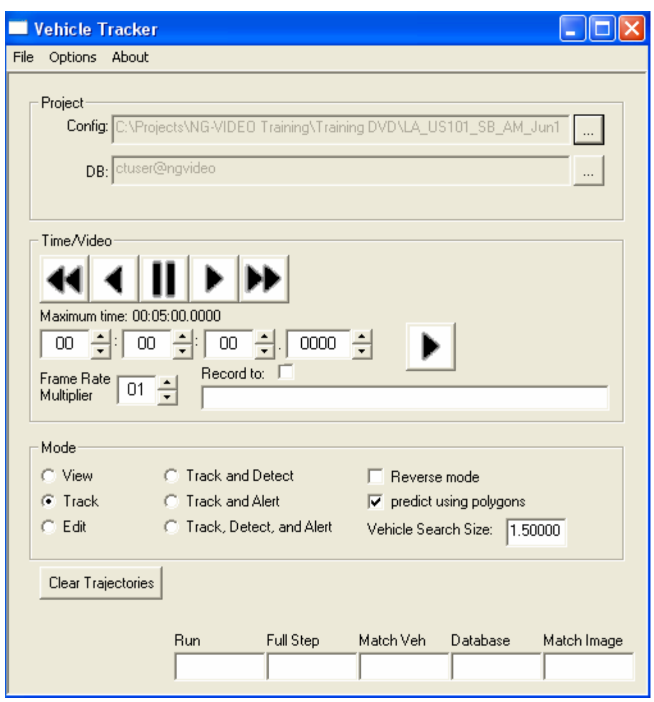
NGSIM-VIDEO has four main windows: 1) Main control panel window; 2) View window; 3) Alert window; and 4) Alert editor window. The functions of these widows are described in this section. Before using NGSIM-VIDEO for tracking purposes, it is necessary to load the configuration file (as discussed in Section 3.0) and connect to the NGSIM-VIDEO database (as discussed in Section 2.0).

4.1 Preparing a Run

NGSIM-VIDEO is built to not require changes to its host computer's registry or other settings. The only requirement for the program is that it be run from its own installation directory. The installation program creates a desktop shortcut that is correctly configured for this to occur. If the shortcut is not on the desktop, a shortcut to the application may be created by right clicking on the NGSIM-VIDEO.exe file and choosing "Create Shortcut." The shortcut will appear at the bottom of the list of files, and the user can drag it onto the desktop for easy access.

To run NGSIM-VIDEO, double click on the desktop shortcut or NGSIM-VIDEO.exe file in the directory when NGSIM-VIDEO is installed. The main control panel will show up on the screen, as shown in Figure 4.1.

Figure 4.1 NGSIM-VIDEO Main Control Panel



NGSIM-VIDEO will not run properly without a proper configuration file and a database. The following steps must be followed before running NGSIM-VIDEO.

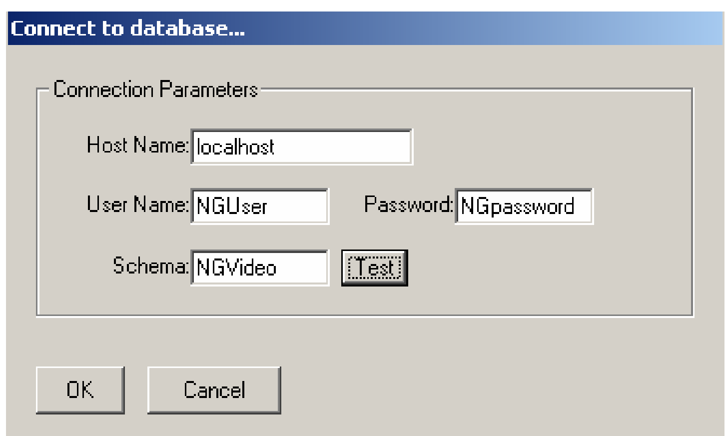
Loading the Configuration File

The first step to prepare for running NGSIM-VIDEO is to load the configuration file. To do this, left click on the File pull-down menu and select "Load Calibration File." This may also be done by left clicking on the button associated with "Config" in Project frame. The users will then be prompted for the path of the configuration file they want to use. This is an xml file, as discussed in Section 3.0. After selecting the OK button, the file path should appear in the upper-most text box of the NGSIM-VIDEO Main Control Panel window.

Connecting to the Database

The second step to prepare NGSIM-VIDEO for running is to connect with the NGSIM-VIDEO database as discussed in Section 2.0. To do this, left click on the File pull-down menu and select "Connect to Database." This may also be done by left clicking on the button associated with "DB" in Project frame. A login window will appear, and the user must input the user name, password, and the name of the database, as illustrated in Figure 4.2.

Figure 4.2 Connect to Database



4.2 MAIN CONTROL PANEL

The Control Panel, shown in Figure 4.1, has control buttons similar to a video cassette recorder (VCR). This panel is used for the main control of the tracking operation.

Main Menu

There are three drop-down menus in the Main Menu, for example: 1) File, 2) Options, and 3) About.

File

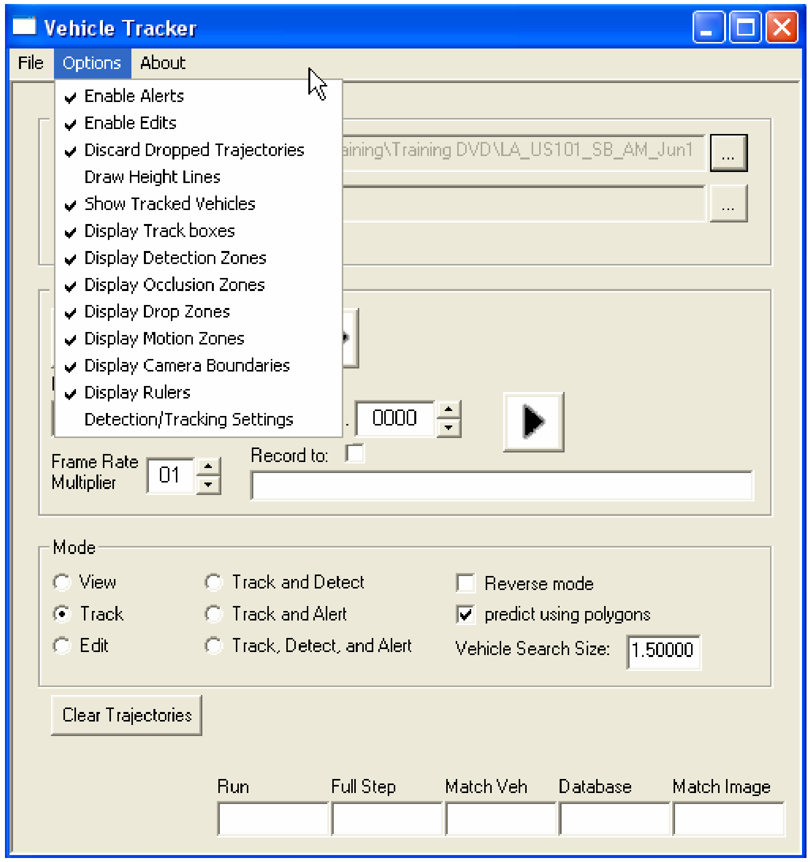
- Load Calibration File: It is used to load the configuration file, as discussed in Section 4.1.
- Connect to Database: It is used to connect to the NGSIM-VIDEO database, as discussed in Section 4.1.
- Close: It closes all current tracking settings, except for the NGSIM-VIDEO software application.
- Exit: It closes all current tracking settings and exits NGSIM-VIDEO application. At the same time, a save file, NGSIM-VIDEO.sav, is created in the same directory when installing NGSIM-VIDEO so that NGSIM-VIDEO will go to the exact time being tracked last.

Options

All of the following items are options available in the Main Menu window Options drop-down menu. The options are activated when a check mark is entered in the drop-down menu list, as illustrated in Figure 4.3. Most options are self-explanatory.

- Enable Alerts: This option activates the automatic alert function and is selected by default.
- Enable Edits: This option allows Edit mode to be activated. The Edit mode is used for trajectory corrections.
- Discard Dropped Trajectories: Currently not available.
- Draw Height Lines: Currently not available.
- Show Tracked Vehicles: Selecting this option creates a visible trajectory line that trails each vehicle being tracked by NGSIM-VIDEO.
- Display Track Boxes: Selecting this option makes the boxes that define the vehicle boundaries appear on the ortho-rectified image View Window. The vehicle identification numbers (VID) are also displayed inside the track boxes.

Figure 4.3 Options Menu in Main Control Panel



- Display Detection Zones: Selecting this option outlines the detection zones in blue in the View window. Deselecting this option will hide the detection zone boxes.
- Display Occlusion Zones: Selecting this option outlines occlusion zones, such as the camera boundary zones where vehicles are moving from one camera to the other. With this option selected, the occlusion zones will be displayed in turquoise in the View window.
- Display Drop Zones: Selecting this option outlines the zones where vehicles are dropped at the end of the study area. The drop zones will be displayed in yellow in the View window.
- Display Motion Zones: Selecting this option outlines the direction polygons as well as the directional points within the polygon.
- Display Camera Boundaries: Selecting this option outlines a line in black between neighboring cameras in the View window.
- Display Rulers: Selecting this option creates horizontal and vertical rulers (in feet) in the View window.

- Detection/Tracking Settings: Selecting this option will direct the user to Vision Algorithm Settings. Users may adjust the parameters to improve detection and tracking. However, it is strongly recommended to keep the default values.

Run Control

Play

Forward play is activated using the right arrow in the Main Menu Control Panel: ▶ This plays the video in a forward continuous stream in the View Window.

Reverse Play

Reverse play is activated using the left arrow in the Main Menu Control Panel: ◀ This plays the video in a backwards continuous stream in the View Window.

Frame Advance

The user can advance frame by frame forward or backward using the following buttons on the Main Menu Control Panel, respectively: ⏩ ⏪ For tracking purposes, it is recommended to use this option frame by frame.

Pause

The pause button is used to pause the video during forward or reverse play modes. This button appears as ⏸ on the Main Menu Control Panel.

Manual Time Input

If the user wants to go to a specific time of the video, double-click inside the hours, minutes, seconds, and decimal seconds boxes, and type in the desired time. Pressing the Enter key or clicking on the right arrow "go" ▶ button, located to the right of the decimal seconds box, displays the frame of video specified. A warning box appears when an out of the range value is specified by the user.

Mode Control

There are eight modes indicated by the radio and check buttons that allow NGSIM-VIDEO to run in different ways. The purpose of each mode is described below.

(1) View

To activate this mode, click on the radio button next to "View" in the main control panel. This mode will play the video only, and will not track or detect vehicles. Therefore, if the user switches to the View mode after tracking some vehicles, NGSIM-VIDEO will stop tracking vehicles, and just play the video image. The tracked information will not be lost. To continue tracking from the point you left off from, simply reset the video to the time you left off, and choose other modes.

(2) Track

To activate this mode, click on the radio button next to "Track" in the main control panel. This mode tracks the vehicles on screen that have already been detected, either automatically or manually, but does not detect vehicles or alert the user of any warnings or fatal errors.

(3) Edit

To activate this mode, click on the radio button next to "Edit" in the main control panel. This mode is useful when vehicles have already been detected and tracked but part of the trajectories needs to be revised.

(4) Track and Detect

To activate this mode, click on the radio button next to "Track and Detect" in the main control panel. This mode will detect and track vehicles, but it will not alert the user of any warnings or fatal errors.

(5) Track and Alert

To activate this mode, click on the radio button next to "Track and Alert" in the main control panel. This mode will detect vehicles and track them from start to end. This mode will also automatically stop the program when a fatal alert appears in the Alert List.

(6) Track, Detect, and Alert

To activate this mode, click on the radio button next to "Track, Detect, and Alert" in the main control panel. This mode will detect vehicles and track them from start to end. This mode will also automatically stop the program when a fatal alert appears in the Alert List.

(7) Reverse Mode

To activate this mode, click on the check button next to "Reverse mode" in the main control panel. When the distance covered by the study area is long, it is suggested to separate the tracking task into two parts, one for forward tracking and the other for reverse tracking. The camera that is looking straight down from a building is usually set as a base camera for both forward and reverse tracking. The reverse tracking video images are reversed so that vehicles appear traveling backward. If tracking is split into two parts, forward tracking has to be completed before activating reverse tracking. For reverse tracking, vehicle information is retrieved from the database generated by the forward tracking.

(8) Predict using polygons

To activate this mode, click on the check button next to "predict using polygons" in the main control panel. By checking this button, NGSIM-VIDEO will predict vehicle moving directions using the direction polygons and directional points in the configuration file (as discussed in Section 3). It is suggested to keep this mode activated, especially on winding roadways or on streets with left- and right- turn movements.

There is a "Vehicle Search Size" appearing right below Predict Using Polygons, which is a parameter used in the vehicle matching algorithm in the NGSIM-VIDEO. The default value is 1.5, as indicated in Figure 4.1. By reducing this number, the time used for vehicle searching and matching might be shortened, and therefore performing tracking might be faster. However, it might also increase the chance of vehicles getting lost in tracking. Therefore, it is recommended to use the default value of 1.5. The NGSIM team has tried 1.2 and it has worked reasonably well.

Clear Trajectories

By hitting the "Clear Trajectories" button in the main control panel, all tracking information will be deleted from the tracking time and after. Users should be careful when using this button. To recover accidentally deleted data, it is strongly suggested that the users back up the NGSIM-VIDEO database regularly.

Notes

There is a status bar at the bottom of the main control panel. It displays the time (in seconds) used for tracking and matching vehicles, searching from database, etc.

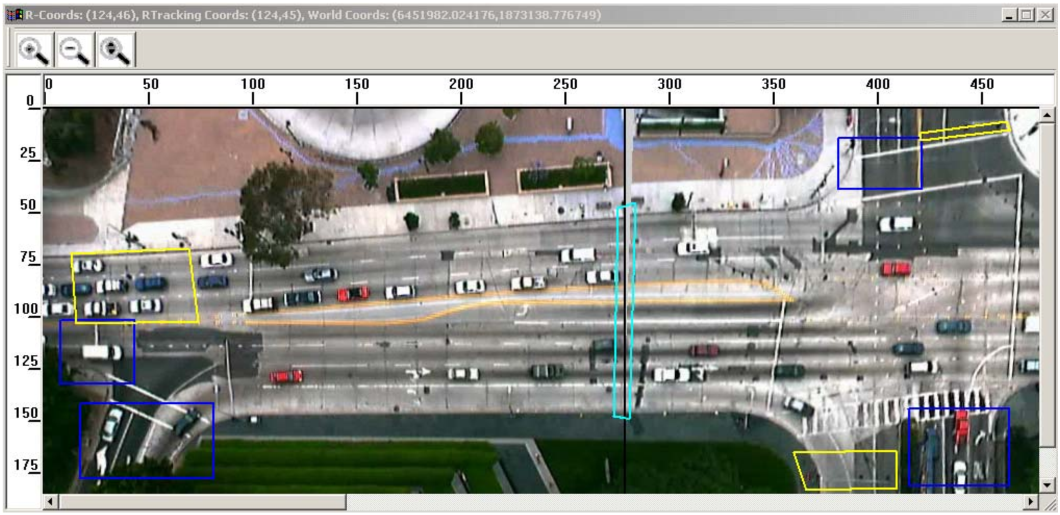
Users can also choose a "frame rate multiplier" in the main control panel. It is usually set to 1, which means that NGSIM-VIDEO tracks vehicles from the video frame by frame without jumping. On the other hand, for example, if the value is set to 10, NGSIM-VIDEO will only track vehicles every 10 frames.

Users can also record the video to a new video by checking "Record to" in the main control panel. This option makes it convenient to obtain a processed video with vehicle tracking boxes on top of the vehicles so that users can check the tracking results visually.

4.3 VIEW WINDOW

The View Window, illustrated in Figure 4.4, shows all camera ortho-rectified images side by side in one window in order for the user to view and track vehicles.

Figure 4.4 View Window



Zoom-in

The user may zoom-in to the video image by pressing this button: 

Zoom-out

The user may zoom-out of the video image by pressing this button: 

Zoom Extents

The user may zoom the video image to its full extent by pressing this button: 

Resize Window

All windows in NGSIM-VIDEO may be resized by clicking on the lower right corner of the window and dragging left, right, up, or down.

Scrollbars

The scrollbars on the horizontal and vertical axes of the View Window can be used just like any Windows scrollbar to pan the image.

Tracking Functions

Edit

To edit the vehicle information, right click on the desired vehicle and select "Edit" from the pop-up menu. Information is displayed about the vehicle such as vehicle ID, vehicle length, vehicle width, vehicle speed, directional zone (or direction polygon as discussed in Section 3.0).

If the user is not satisfied with the vehicle length or vehicle width of the tracking box, the Edit function serves as an alternative to change those values.

As discussed in Section 3.0, there could be overlapping direction polygons (zones) such as a shared lane with through and right-turn movements. The Edit function allows the user to choose the right direction polygon from the drop-down list. The selected direction polygon as well as directional points will be highlighted in red in the View Window.

Delete

To completely delete a vehicle's trajectory from the database, right click on the desired vehicle and select "Delete" from the pop-up menu. The vehicle trajectory will disappear from the View Window and the vehicle will no longer be tracked. In order to track the vehicle with a new vehicle identification number (VID) from that point forward, the user must left click on the upper left (UL) of the vehicle once, and then move the arrow to the upper right (UR) of the vehicle and click again. These two clicks together with the default vehicle width (as defined in the configuration file) create a rectangle over the vehicle.

Truncate

To go back to a certain frame and start re-tracking a particular vehicle from that point in time, right click on the vehicle and select "truncate" from the pop-up menu.

Manual Alert

To create an alert manually for a vehicle, right click on it and select "Manual Alert" from the pop-up menu. This will allow you to manually correct vehicle trajectories in the Alert Editor window.

Custom Alert

This option is currently not being used.

Notes

In order to allow NGSIM-VIDEO to track a vehicle that is not detected automatically, the user may choose to track the vehicle by manually selecting it. To do this, the user should left click on the upper left (UL) corner of the vehicle image once, and then move the arrow to the upper right (UR) corner of the vehicle image and click again. These two clicks together with the default vehicle width (as defined in the configuration file) create a rectangle over the vehicle and tells NGSIM-VIDEO where to look for a vehicle. It also assigns a VID to that vehicle in sequential order.

4.4 ALERT LIST WINDOW

All alerts generated by NGSIM-VIDEO will appear in this window. Clicking on any alert will cause the video of the particular vehicle in question to appear in the Alert Editor Window at the frame in which the alert occurred. The View Window will remain at the point at which the NGSIM-VIDEO tracker stopped.

Vehicle Identification Number (VID)

The VID number appears in the first column of the Alert List. This number is assigned by NGSIM-VIDEO at the point the vehicle is detected and tracking has started.

Alert Time

This is the time when the alert is generated.

Last Time

This is the time when vehicle tracking incurs an error.

Last Y

This is the y-coordinate of the vehicle causing the alert.

Type

This is the severity of the warning generated by NGSIM-VIDEO.

Comments

The user can enter comments for each alert by clicking in the text boxes of the respective alerts and typing notes.

4.5 ALERT EDITOR WINDOW

This window is used to edit trajectories when errors occur.

Alert Type

There are four alert types in NGSIM-VIDEO. They are described below.

Drop Alert

A drop alert is generated by NGSIM-VIDEO when a vehicle's trajectory does not follow the vehicle for five consecutive frames. This is considered a fatal error, and therefore the video is paused and the user is allowed to edit the trajectory.

Overlap Alert

This type of alert is generated when two or more vehicles' track boxes overlap. Generally, if the user is creating track boxes for any vehicle, it should be within the visible confines of the vehicle so that overlap alerts are minimized.

Drift Alert

A drift alert is created when a vehicle tracking box veers to the left or right of the last direction vector in any one time step by more than a user-specified angle. The default setting is 15 degrees for a warning level alert and 30 degrees for an exception (fatal) level alert.

Stop Alert

This type of alert is generated whenever a vehicle is determined to have stopped for an unusual amount of time. Users may set NGSIM-VIDEO to ignore these alerts when there are occasional vehicle stops such as on congested freeways and on arterials with traffic signals.

Setting Alert Levels

Alert levels for drop, overlap, drift, and stop may be set by selecting "Alert Level Settings" from the "Options" pull-down menu in the Alert Editor Window.

Warning Level

The warning alert occurs whenever the lower bound of a set value is exceeded when tracking a vehicle. NGSIM-VIDEO keeps running, and a description including the VID, alert type, and alert time appears in the Alert List Window. However, nothing appears in the Alert Editor Window.

Exception Level

The exception (fatal) alert occurs whenever the upper bound of a set value is exceeded. The video will pause and the vehicle causing the fatal alert will be highlighted in the Alert Editor Window with the track box shown.

Ignore Automatic Alert

Automatic alerts can be ignored for each alert type by selecting "Ignore" from "Alert Level Settings" which can be accessed from the "Options" pull-down menu in the Alert Editor Window.

Trajectory Correction

Users can manually correct trajectories using the Alert Editor Window. To do this, click the alert on the Alert List window or create a manual alert by right clicking on the vehicle tracking box in View Window. In the Alert Editor Window, go back to a time at which the vehicle position is correct, and click "Set

Initial Time" button. Move forward to a time, which may be beyond the current time being tracked, click on the vehicle's tracking box, move to the center of the vehicle as it appears in the Alert Editor Window, left click one time, and click "Fixed" to quit the Alert Editor Window. This creates a new tracking trajectory for the vehicle and allows continuing the tracking process. After this process, tracking can be continued by pressing forward play on the Main Menu Control Panel.

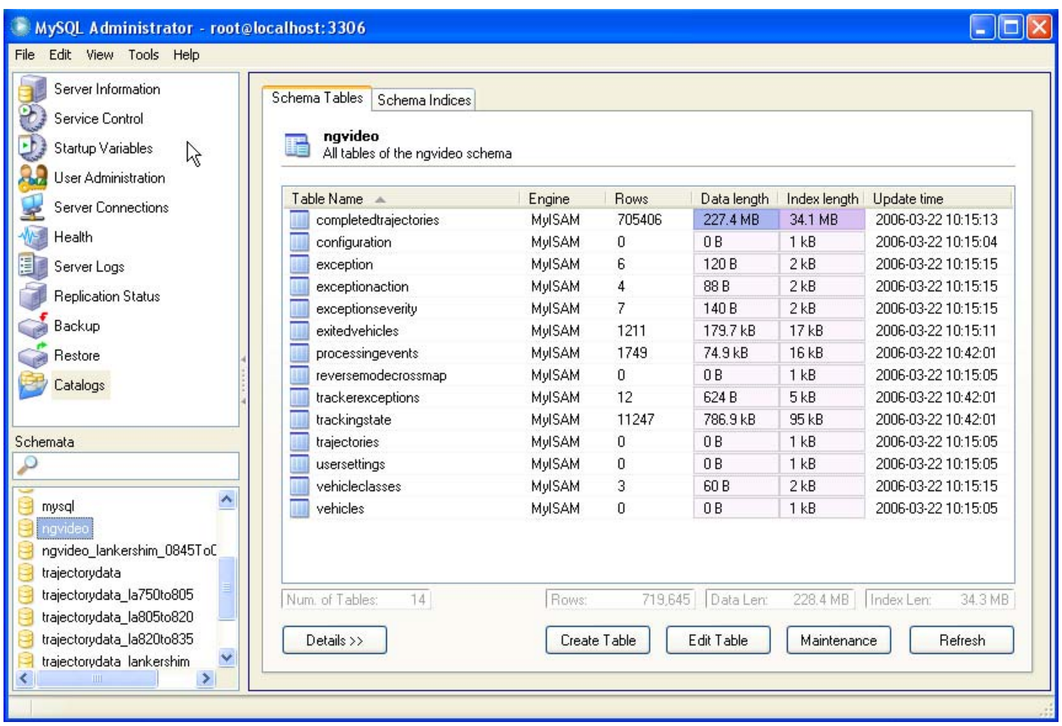
5.0 NGSIM-VIDEO Database

The NGSIM-VIDEO database serves as the repository of the output data from NGSIM-VIDEO tracking procedures. Section 2.0 covered how to install MySQL and NGSIM-VIDEO database. In this section, key tables and fields are discussed. A few useful commands to manipulate the database are also discussed.

5.1 Understanding NGSIM-VIDEO Database

There are 14 tables in the NGSIM-VIDEO database, as illustrated in Figure 5.1 (in the right-side pane). Among them, *completedtrajectories*, *trajectories*, *exitedvehicles*, and *vehicles* are four key tables. When a vehicle is being tracked, the vehicle-related information is stored in *trajectories* table and *vehicles* table. Once the vehicle exits the tracking system by reaching one of the drop zones, the vehicle-related information will be automatically transferred from *trajectories* table to *completedtrajectories* table, and from *vehicles* table to *exitedvehicles* table. Fields in these tables are detailed below.

Figure 5.1 Tables in NGSIM-VIDEO Database



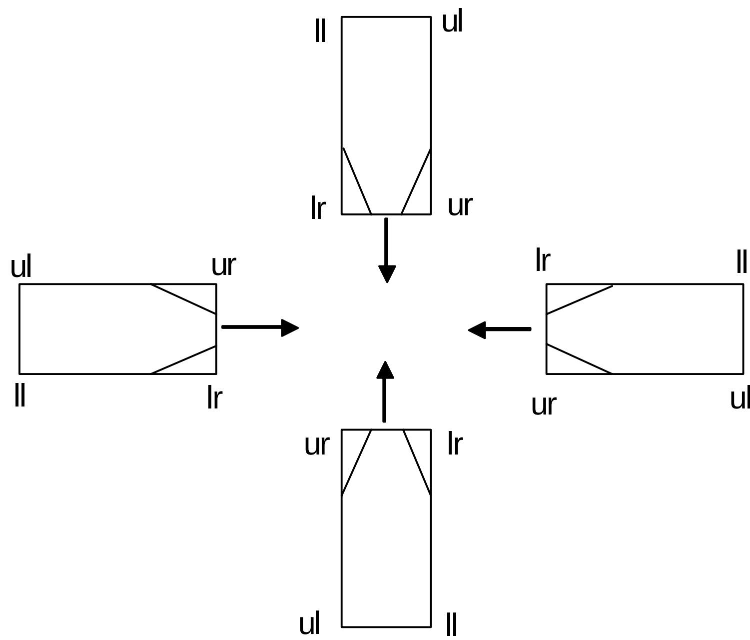
Completedtrajectories Table

This table is used to store trajectory information for tracked vehicles. There are 46 fields defined in this table. Some of the fields are either defined for program internal use or for future program expansions. Only those fields which are directly related to trajectory outputs are described herein, as listed in Table 5.1. The four corner points of the vehicle being tracked, i.e., upper left (UL), upper right (UR), lower right (LR), and lower left (LL), are illustrated in Figure 5.2.

Table 5.1 Fields in Completed Trajectories Table

Field	Description
Vid	Vehicle identification number, which is assigned by NGSIM-VIDEO sequentially when the vehicle is detected and tracking is started
Time	Time step when the vehicle is tracked. If 10 frames per second is used, divide the value by (10*1,000). For example, 117,000 means 117,000/(10*1,000) = 11.7 seconds.
cameraNumber	Camera number from the left to the right. It starts from 0.
isreversemode	It is equal to 0 for forward tracking; otherwise, it's 1 for reverse tracking.
ulImageX	X coordinate of the upper left (UL) corner of the vehicle, as illustrated in Figure 5.2, on the video image in pixels
ulImageY	Y coordinate of the upper left (UL) corner of the vehicle on the video image in pixels
urImageX	X coordinate of the upper right (UR) corner of the vehicle, as illustrated in Figure 5.2, on the video image in pixels
urImageY	Y coordinate of the upper right (UR) corner of the vehicle on the video image in pixels
lrImageX	X coordinate of the lower right (LR) corner of the vehicle, as illustrated in Figure 5.2, on the video image in pixels
lrImageY	Y coordinate of the lower right (LR) corner of the vehicle on the video image in pixels
llImageX	X coordinate of the lower left (LL) corner of the vehicle, as illustrated in Figure 5.2, on the video image in pixels
llImageY	Y coordinate of the lower left (LL) corner of the vehicle on the video image in pixels
ulX	X world coordinate, in feet, of the upper left (UL) corner of the vehicle, as illustrated in Figure 5.2.
ulY	Y world coordinate, in feet, of the upper left (UL) corner of the vehicle.
urX	X world coordinate, in feet, of the upper right (UR) corner of the vehicle, as illustrated in Figure 5.2.
urY	Y world coordinate, in feet, of the upper right (UR) corner of the vehicle.
lrX	X world coordinate, in feet, of the lower right (LR) corner of the vehicle, as illustrated in Figure 5.2.
lrY	Y world coordinate, in feet, of the lower right (LR) corner of the vehicle.
llX	X world coordinate, in feet, of the lower left (LL) corner of the vehicle, as illustrated in Figure 5.2.
llY	Y world coordinate, in feet, of the lower left (LL) corner of the vehicle.

Figure 5.2 Definition of Vehicle Corner Points: UL, UR, LR, and LL



Trajectories Table

The *trajectories* table has similar data fields as in the *completedtrajectories* table. As discussed earlier, the fields in the *trajectories* table are transferred automatically to the *completedtrajectories* table once the vehicle exits the tracking system.

Exitedvehicles Table

This table is used to store vehicle-related information for those vehicles which have already exited the tracking system. There are 49 fields defined in *exitedvehicles* table. Some of the fields are either defined for program internal use or for future program expansions. Only those fields which are directly related to trajectory outputs are described herein, as listed in Table 5.2.

Table 5.2 Fields in Exited Vehicles Table

Field	Description
Vid	Vehicle identification number, which is assigned by NGSIM-VIDEO sequentially when the vehicle is detected and tracking is started
time_first	Time step when the vehicle first enters the tracking system. If 10 frames per second is used, divide the value by (10*1,000). For example, 132,000 means 132,000/(10*1,000) = 13.2 seconds.
time_last	Time step when the vehicle exits the tracking system. If 10 frames per second is used, divide the value by (10*1,000). For example, 237,000 means 237,000/(10*1,000) = 23.7 seconds.
Isreversemode	It is equal to 0 for forward tracking; otherwise, it's 1 for reverse tracking.
ulImageX_first	X coordinate of the upper left (UL) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
ulImageY_first	Y coordinate of the upper left (UL) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
urImageX_first	X coordinate of the upper right (UR) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
urImageY_first	Y coordinate of the upper right (UR) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
lrImageX_first	X coordinate of the lower right (LR) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
lrImageY_first	Y coordinate of the lower right (LR) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
llImageX_first	X coordinate of the lower left (LL) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
llImageY_first	Y coordinate of the lower left (LL) corner of the vehicle when the vehicle first enters the tracking system on the video image in pixels
ulX_first	X world coordinate, in feet, of the upper left (UL) corner of the vehicle when the vehicle first enters the tracking system.
ulY_first	Y world coordinate, in feet, of the upper left (UL) corner of the vehicle when the vehicle first enters the tracking system.
urX_first	X world coordinate, in feet, of the upper right (UR) corner of the vehicle when the vehicle first enters the tracking system.
urY_first	Y world coordinate, in feet, of the upper right (UR) corner of the vehicle when the vehicle first enters the tracking system.
lrX_first	X world coordinate, in feet, of the lower right (LR) corner of the vehicle when the vehicle first enters the tracking system.
lrY_first	Y world coordinate, in feet, of the lower right (LR) corner of the vehicle when the vehicle first enters the tracking system.
llX_first	X world coordinate, in feet, of the lower left (LL) corner of the vehicle when the vehicle first enters the tracking system.
llY_first	Y world coordinate, in feet, of the lower left (LL) corner of the vehicle when the vehicle first enters the tracking system.

Vehicles Table

The *vehicles* table has similar data fields as in the *exitedvehicles* table. As discussed earlier, the fields in the *vehicles* table are transferred automatically to the *exitedvehicles* table once the vehicle exits the tracking system.

Notes

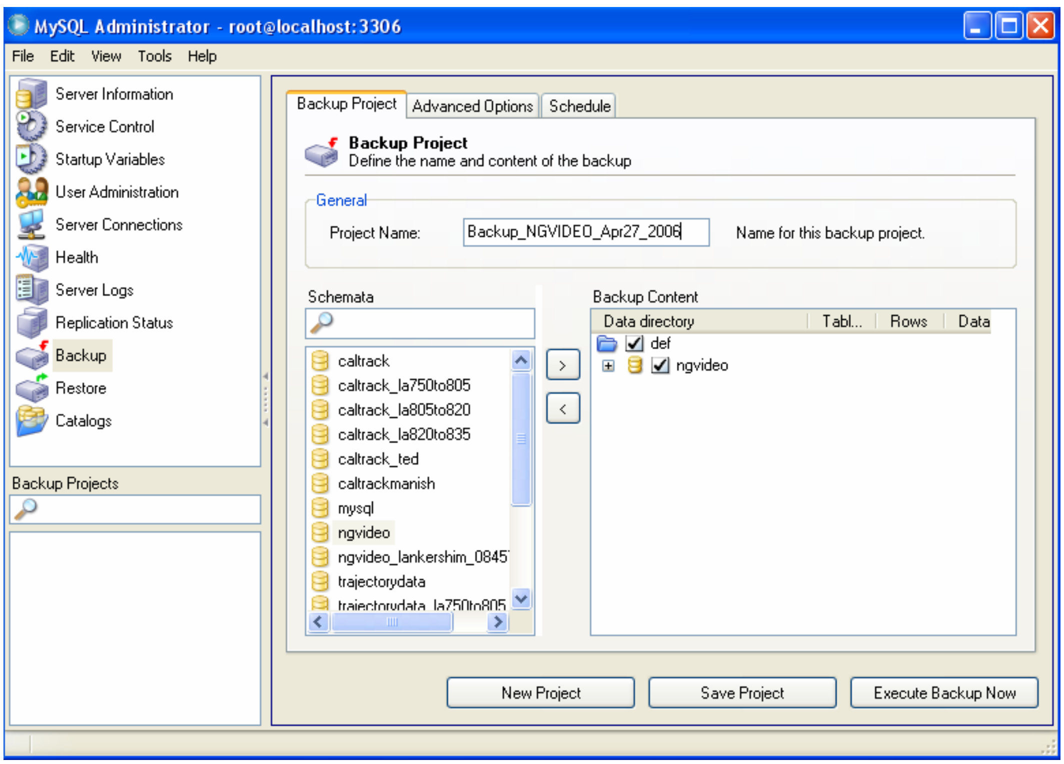
The *completedtrajectory* table contains all useful trajectory data. Therefore, users only have to consider *completedtrajectory* table when post-processing the data.

5.2 Manipulating the NGSIM-VIDEO Database

Backup NGSIM-VIDEO Database

It is strongly recommended that users backup the NGSIM-VIDEO database frequently. MySQL Administrator can be used for backup. Simply open MySQL Administrator and create a New Project for the backup by providing the backup file name and the database, as illustrated in Figure 5.3.

Figure 5.3 Backup NGSIM-VIDEO Database

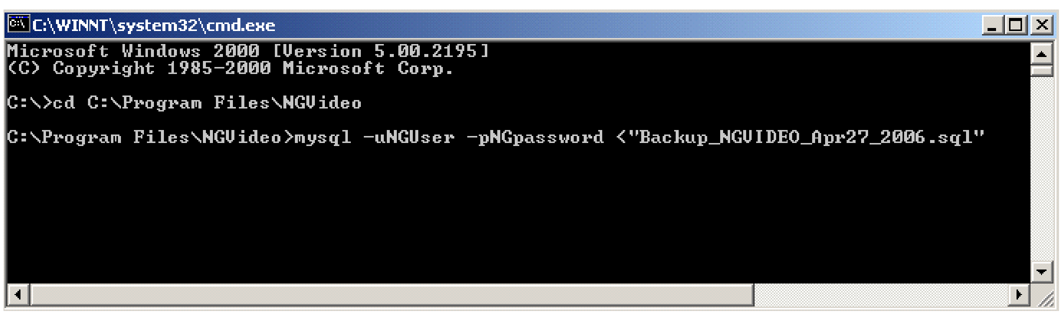


Restore NGSIM-VIDEO Database

As discussed in Section 2.0, MySQL Administrator can be used to restore MySQL database. However, it is noted that MySQL Administrator may not correctly restore databases especially when the size of the file is large, which is usually the case when restoration from a previous backup file is necessary. It is recommended to use the restore command in MS-DOS (Microsoft – Disk Operating System) environment, instead. The steps are illustrated in Figure 5.4 and are detailed below:

- Access MS-DOS from Microsoft Windows by clicking Start/Run and typing "**command**" or by typing "**CMD**" in Windows 98, Windows NT, Windows 2000 or Windows XP.
- Go to the directory that contains the database file for restoration, for example, C:\Program Files\NGSIM-VIDEO\Backup_NGVIDEO_Apr27_2006.sql. Type "cd C:\Program Files\NGSIM-VIDEO" and hit the enter key to change directory (cd) to C:\Program Files\NGSIM-VIDEO where the backup file is located.
- Under the current prompt, type: mysql –u*user* –p*password* < "Backup_NGVIDEO_Apr27_2006.sql" and hit the enter key.

Figure 5.4 Restore Database Using MS-DOS Command



Basic Commands to Manipulate the NGSIM-VIDEO Database

Some basic commands to manipulate the NGSIM-VIDEO database are discussed below. Users are advised to review the MySQL manual for more detailed coverage.

Connect to MySQL Server

To connect to the MySQL server, the user is required to provide a MySQL username and a password. If the server runs on a machine other than the one where the user is logged in, the user also need to specify a hostname. The following command is used to connect:

C:\>mysql -h *host* -u *user* -p

Enter password: ********

host represents the hostname (for example, localhost) where the MySQL server is running, and *user* represents the username (for example, root) of your MySQL account. Substitute appropriate values for your setup. The ******** represents your password; enter it when MySQL displays the "Enter password" prompt.

For efficiency, MySQL should be installed on the local computer for using NGSIM-VIDEO. In this case, the user can ignore the "host" part and connect to the MYSQL server with the following command:

C:\>mysql -u *user* -p

Enter password: ********

Disconnect MySQL Server

The user can disconnect at any time by typing EXIT or QUIT at the mysql> prompt:

mysql> QUIT (or mysql> EXIT)

Find Databases

Use the SHOW statement to find out what databases currently exist on the server:

mysql> SHOW DATABASES;

Access a Database

If the database exists, try to access it:

mysql> USE *DatabaseName*;

For example,

mysql> USE ngvideo;

Show Tables in Database

The number of tables included in the database can be checked using the SHOW statement:

mysql> SHOW TABLES;

Select All Data From a Table

Use the SELECT statement to retrieve everything from a table by using "*" in the statement. This form of SELECT is useful if you want to review your entire table.

mysql> SELECT * FROM *TableName*;

Select Particular Rows From a Table

The user can select only particular rows from a table. For example, if the user wants to verify if the vehicle with VID of 75 has been tracked successfully, that vehicle's record may be selected using the following command:

```
mysql> SELECT * FROM completedtrajectories WHERE vid = 75;
```

Select Particular Columns From a Table

If the user does not want to see all the columns in a table, s/he may name specific columns, separated by commas. For example, if the user wants to know VID and time from *completedtrajectories* table, select the *vid* and *time* columns:

```
mysql> SELECT vid, time FROM completedtrajectories;
```

Select Particular Rows and Columns From a Table

The user can select only particular rows and particular columns from a table. For example, if the user wants to know VID and time from *completedtrajectories* table for the vehicle with VID of 75, use the following SELECT statement:

```
mysql> SELECT vid, time FROM completedtrajectories WHERE vid = 75;
```

Sort Rows

It is often easier to examine query output when the rows are sorted. To sort a result, use an ORDER BY clause. For example,

```
mysql> SELECT vid, time FROM completedtrajectories ORDER BY time;
```

The default sort order is ascending, with smallest values first. To sort in descending order, add the DESC keyword to the name of the column you are sorting by. For example,

```
mysql> SELECT vid, time FROM completedtrajectories ORDER BY time
DESC;
```

Count Rows

It is often wanted to know how many rows are available. COUNT(*) is used to count the number of rows. For example, the query to count how many vehicle trajectory lines in the *completedtrajectories* table looks like this:

```
mysql> SELECT COUNT(*) FROM completedtrajectories;
```

If the user wants to know the number of trajectory lines for a specific vehicle (e.g., VID = 75), use a WHERE clause.

```
mysql> SELECT COUNT(*) FROM completedtrajectories WHERE
vid = 75;
```

If the user wants to know how many distinct vehicles are in the *completedtrajectories* table, use a DISTINCT clause.

mysql> SELECT COUNT(DISTINCT vid) FROM completedtrajectories;

Output to Text File

Data in MySQL tables can be outputted to a text file by using SELECT INTO OUTFILE statement. For example, the user can save all the data from the *completedtrajectories* table into a text file, as shown below.

mysql>SELECT * INTO OUTFILE "completedtrajectories.txt" FROM completedtrajectories;

Consistency Check

In the NGSIM-VIDEO database, the number of distinct vehicles in the *completedtrajectories* table and in the *exitedvehicles* table should be equal. Users can check these using the following SELECT statements:

mysql> SELECT COUNT(DISTINCT vid) FROM completedtrajectories; and

mysql> SELECT COUNT(DISTINCT vid) FROM exitedvehicles.

The number of distinct vehicles in the *trajectories* table and in the *vehicles* table should also be equal.

mysql> SELECT COUNT(DISTINCT vid) FROM trajectories; and

mysql> SELECT COUNT(DISTINCT vid) FROM vehicles.

6.0 Examples

NGSIM-VIDEO is an open source software application distributed by the Federal Highway Administration (FHWA). It is a program developed by Cambridge Systematics, Inc. through the Next Generation Simulation (NGSIM)[15] project. The objective of NGSIM is to develop a core of open behavioral algorithms in support of microscopic traffic simulation, with supporting documentation and validation data sets.

Two of the data collection and tracking efforts are described in this section: 1) U.S. 101 Freeway in Los Angeles, California; and 2) Lankershim Boulevard in Universal City, California.

6.1 EXAMPLE 1: U.S. 101 FREEWAY

Video and trajectory data were collected on a segment of U.S. 101 (Hollywood Freeway) in Los Angeles, California on June 15, 2005. The NGSIM-VIDEO program was used to transcribe 45 minutes of vehicle trajectory data from the videos. The following section describes the processes used to collect and subsequently process this data using the NGSIM-VIDEO program.

Study Area Description

Digital videos of traffic movements were collected using eight video cameras temporarily mounted on the rooftop of a 36-story office building, at 10 Universal City Plaza (UCP), located immediately adjacent to the U.S. 101 and Lankershim Boulevard interchange in Universal City, California. Video data for the northbound and the southbound directions were captured on subsequent weekdays in June 2005; however, only a subset of the southbound data has been processed to date.

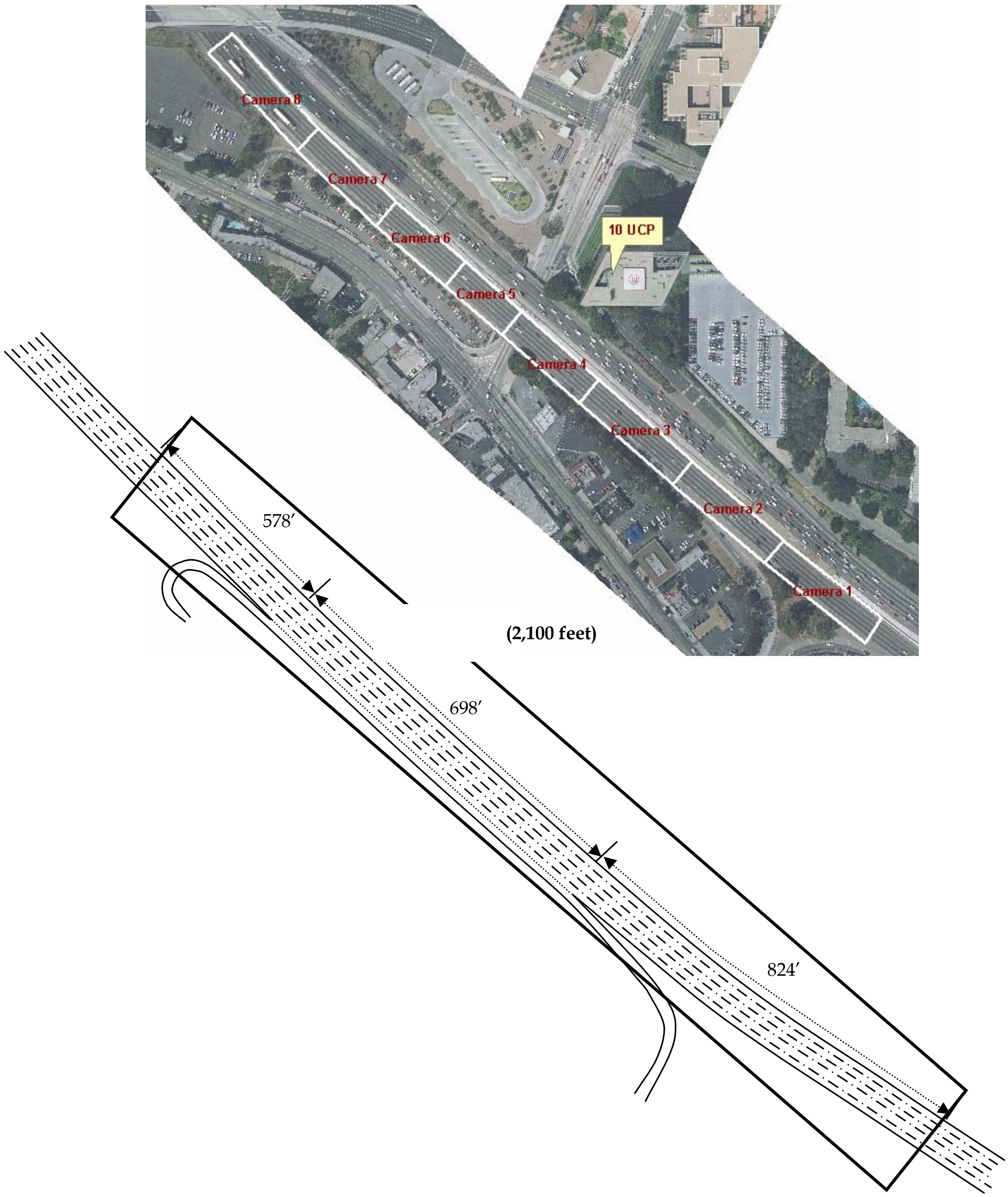
Figure 6.1 provides a schematic illustration of the southbound study area. The site was approximately 2,100 feet in length, with five mainline lanes throughout the section. An auxiliary lane is present through a portion of the corridor between the on-ramp at Ventura Boulevard and the off-ramp at Cahuenga Boulevard.

This particular data collection site was selected based on the geometrics of the freeway weave section, the congestion characteristics on the corridor, the height of the building., the proximity of the building to the study corridor, and the lack

[15] http://ngsim.fhwa.dot.gov/.

of significant visual occlusions from the vantage point (e.g., trees, overpasses, overhead signs, etc.)

Figure 6.1 Study Area Schematic and Camera Coverage



Video Collection

Video data were collected using eight SONY DFW-VL500 video cameras, Cameras 1 through 8, with Camera 1 recording the southernmost, and Camera 8 recording the northernmost section of the study area, as shown in Figure 6.1. These digital cameras are capable of recording video in VGA (640×480) resolution format. The field of view captured by each camera was carefully planned prior to data collection to ensure overlaps of views of at least 80 feet and that the resolution for any particular camera did not fall below one pixel per foot of ground coverage.

Three computers, two connected to three cameras each and the other connected to two cameras, were used to control and store the digital videos. All cameras were synchronized using an automatic synchronization trigger specifically developed for this data collection platform and capable of simultaneously triggering the initiation of data recording from all cameras within 2/1,000th of a second. Digital videos were collected over an approximate nine-hour period from 7:00 a.m. to 12:00 p.m. and from 3:00 p.m. to 7:00 p.m.

During the data collection, supplemental data including freeway loop detector data, ramp meter signal timings, and roadway incident reports were monitored and archived. Additionally, all camera parameters (e.g., zoom setting, focus length, etc.) used for each individual camera were each carefully documented.

NGSIM-VIDEO Preprocessing

Video data representing the southbound direction of U.S. 101 were processed using NGSIM-VIDEO to obtain trajectory data. A total of 45 minutes of trajectory data (from 7:50 a.m. to 8:35 a.m.) were transcribed at a resolution of 10 frames per second. The data were divided into three 15-minute periods (7:50 a.m. to 8:05 a.m., 8:05 a.m. to 8:20 a.m., and 8:20 a.m. to 8:35 a.m.) for processing and analysis.

The processes used to prepare the video files for processing with NGSIM-VIDEO are described in the subsequent sections.

Video Stabilization

In all eight video files, only one was found requiring stabilization. The SteadyHand program was used for the stabilization purpose.

Video Rectification

All eight cameras were calibrated using the same settings recorded during field video collection. All video files were rectified by following the procedure as discussed in Section 3.0. Figure 6.2 shows a snapshot of the source AVI from Camera 4 with the region-of-interest (ROI). As illustrated in Figure 6.3, the ROI in the source AVI was extracted and interpolated to provide a rectified AVI file. As shown in Figure 6.2, the source AVI was recorded with vehicles moving from right to left. For convenience of tracking, the video images were rotated during the rectification process so that all vehicles appeared to be moving from left to right.

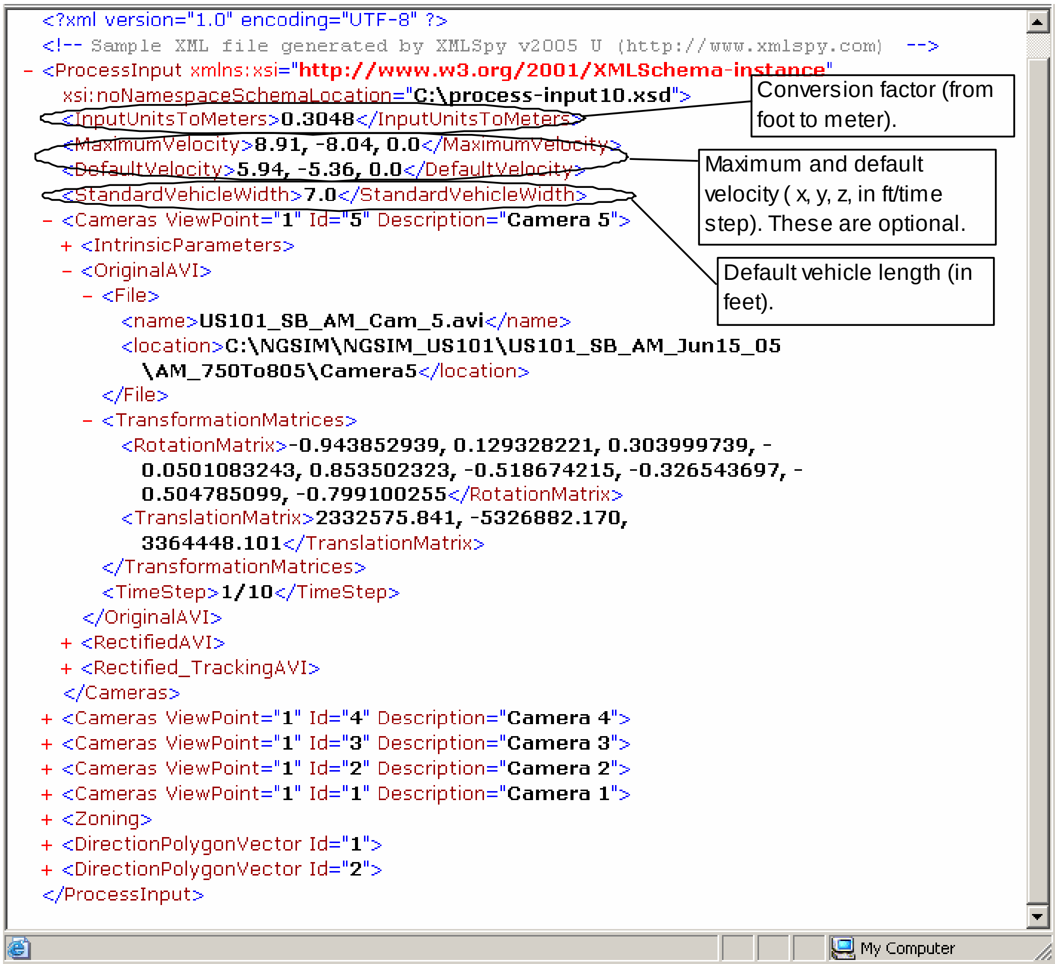
Figure 6.2 Source AVI with the ROI (Camera 4)

Figure 6.3 Rectified AVI of the ROI (Camera 4)

Configuration File

As discussed in Section 3.0, the configuration file is in XML format and includes AVI file names and locations, camera's intrinsic parameters and transformation matrices, input parameters, zoning, and direction polygon with directional points. An overview of the configuration file for U.S. 101 is illustrated in Figure 6.4.

Figure 6.4 Configuration File for U.S. 101



Vehicle Tracking

For each camera, the configuration file has three AVI settings (i.e., stabilized original AVI, rectified AVI, and rectified+tracking AVI). The NGSIM team found that it was not necessary to have separate settings for rectified and rectified+tracking AVIs. Therefore, users can simply copy and paste for the rectified+tracking AVI settings. The redundant structure kept herein is for meeting NGSIM-VIDEO input requirements. A snapshot of settings for the rectified AVI is shown in Figure 6.5. To review other parts of the configuration file, please browse through the file which is included in the NGSIM-VIDEO installation package.

Figure 6.5 Rectified AVI Settings for U.S. 101

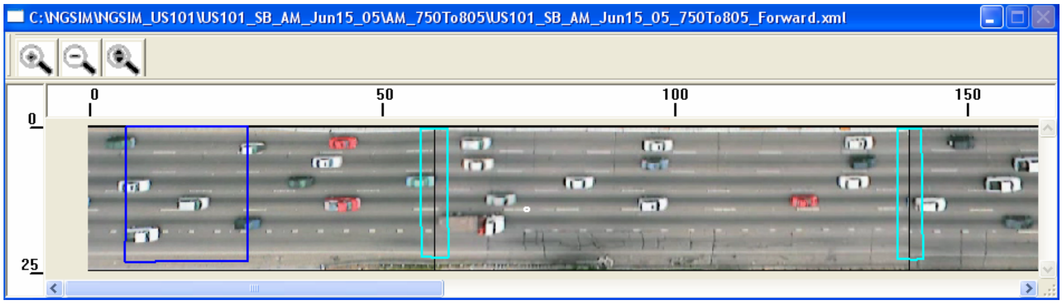

```
- <RectifiedAVI>
  - <File>
      <name>US101_SB_AM_Cam_5_Rectified.avi</name>
      <location>C:\NGSIM\NGSIM_US101\US101_SB_AM_Jun15_05\AM_750To805\Camera5</location>
    </File>
  - <TransformationMatrices>
      <RotationMatrix>0.74923888040, -0.66229985664, 0.00000000000, 0.66229985664, 0.74923888040,
        0.00000000000, 0.00000000000, 0.00000000000, 1.00000000000</RotationMatrix>
      <TranslationMatrix>6451698.59000, 1872837.62000, 1.00000000000</TranslationMatrix>
      <PixelsPerFoot>2.0000</PixelsPerFoot>
    - <WorldBoundaryCoordinates>
      - <Point_3D>
          <X>6451698.5900000</X>
          <Y>1872837.6200000</Y>
          <Z>0.0</Z>
        </Point_3D>
      - <Point_3D>
          <X>6451843.9430247</X>
          <Y>1872709.1345992</Y>
          <Z>0.0</Z>
        </Point_3D>
      - <Point_3D>
          <X>6451789.6347625</X>
          <Y>1872647.6967228</Y>
          <Z>0.0</Z>
        </Point_3D>
      - <Point_3D>
          <X>6451644.2817378</X>
          <Y>1872776.1821236</Y>
          <Z>0.0</Z>
        </Point_3D>
      </WorldBoundaryCoordinates>
    </TransformationMatrices>
    <TimeStep>1/10</TimeStep>
  </RectifiedAVI>
```


NGSIM-VIDEO Processing

The distance covered by the study area is relatively long; therefore, the tracking task was separated into two parts, one for forward tracking and the other for reverse tracking. Camera 5 was set as the base camera for both forward and reverse tracking. Hence, vehicle tracking progress was divided into two major parts: 1) forward (Cameras 5, 4, 3, 2, and 1); and 2) reverse (Cameras 5, 6, 7, and 8). Forward tracking was completed first for all five cameras before reverse tracking. For reverse tracking, no vehicle detection was involved since NGSIM-VIDEO retrieves vehicle location information from the database generated by the forward tracking.

Once the configuration file is loaded and the NGSIM-VIDEO database is connected, NGSIM-VIDEO will launch four windows as discussed in Section 4.0: 1) main control panel, 2) view, 3) alert list, and 4) alert editor windows. Figure 6.6 illustrates the view window for U.S. 101 forward tracking.

Figure 6.6 NGSIM-VIDEO View Window for U.S. 101 Data (Partial View)



NGSIM-VIDEO Database

Following the operating procedures described in Section 4.0 for using NGSIM-VIDEO, a database containing the trajectory information was developed. To provide an example of the magnitude of this dataset, for the time period of 7:50 a.m. to 8:05 a.m., there were a total number of 2,169 vehicles tracked, which resulted in about 1.2 million trajectory data points.

Notes

- When using NGSIM-VIDEO for freeway tracking, direction polygons are not required if the freeway is straight. However, it is recommended to include direction polygons as well as associated directional points in the configuration file, when there is a curve in the freeway study area.
- If a vehicle is not detected automatically, the user can manually detect it. To do this, the user must left click on the upper left (UL, as defined in Figure 5.2) of the vehicle once, and then move the arrow to the upper right (UR, as defined in Figure 5.2) of the vehicle and click again. These two clicks together with the default vehicle width (as defined in the configuration file) create a rectangle over the vehicle and tells NGSIM-VIDEO where to look for a vehicle. It also assigns a VID to that vehicle in sequential order.

6.2 Example 2: Lankershim Boulevard Arterial

Video and trajectory data were collected on a segment of Lankershim Boulevard located near the interchange with U.S. 101 (Hollywood Freeway) in Universal City, California. The NGSIM-VIDEO program was subsequently used to transcribe 30 minutes of vehicle trajectory data from the videos. The following section describes the processes used to collect and process these data using the NGSIM-VIDEO program.

Study Area Description

The videos were collected using five video cameras mounted on a 36-story building, 10 Universal City Plaza (UCP), which is located adjacent to the U.S. 101 and Lankershim Boulevard interchange in Universal City, California.

Figure 6.7 provides an aerial image of the location with the camera coverage. Figure 6.8 presents a schematic illustration of the study area. The site was approximately 1,600 feet in length, with four signalized intersections and three to four arterial through lanes in each direction.

Video Collection

This arterial data collection was completed in coordination with the freeway data collection on U.S. 101 described in Section 6.1. An extra day was used to reposition and configure the data collection platform to collect the arterial data.

The data collection equipment and procedures used were identical to those described in Section 6.1, with the following exceptions. Due to the presence of the Highway 101 overpass, which obscures a significant section of the arterial roadway near the northwest corner of the building, the cameras were positioned to only record a section of the roadway extending from the building to northeast. No attempt was made to record video of the roadway on the opposite side of the freeway overpass. Thus, only five cameras were required to capture the data versus the eight that were required to collect the freeway data. Also, both directions of the arterial movements were captured simultaneously, while the freeway data collection effort recorded the northbound and southbound directions on subsequent days.

Video data were collected using five video cameras, Cameras 1 through 5, with Camera 1 recording the southernmost and Camera 5 recording the northernmost section of the study area, as shown in Figure 6.7. All cameras were synchronized using an automatic synchronization trigger. Digital video images were collected over an approximate 9-hour period from 7:00 a.m. to 12:00 p.m. and from 3:00 p.m. to 7:00 p.m. on June 16, 2005.

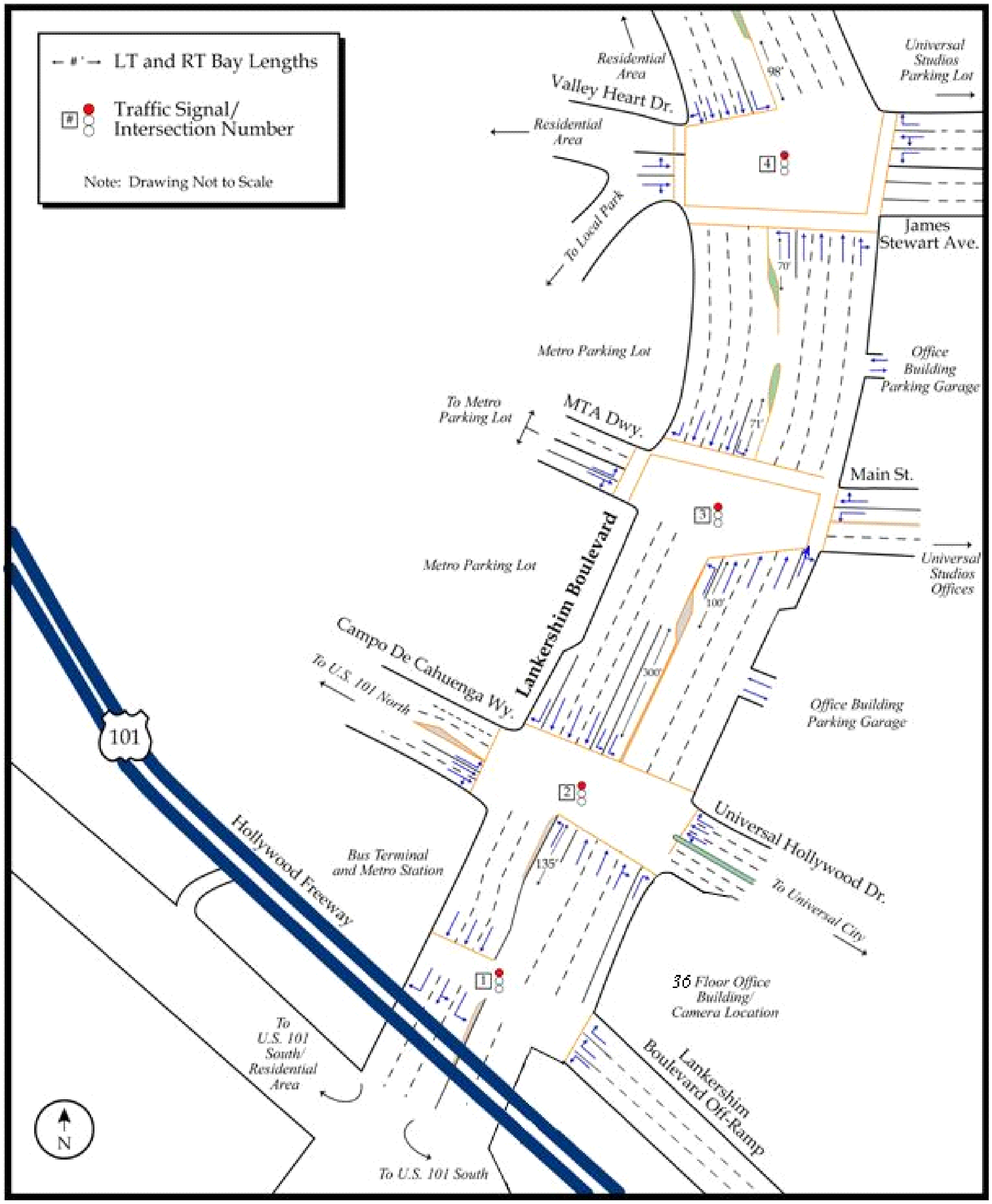
NGSIM-VIDEO Preprocessing

NGSIM-VIDEO was used to obtain trajectory data in the Lankershim Boulevard study area. A total of 32 minutes of trajectory data (from 8:28 a.m. to 9:00 a.m.) were transcribed at a resolution of 10 frames per second. The data were divided into two time periods (8:28 a.m. to 8:45 a.m. and 8:45 a.m. to 9:00 a.m.) for processing and analysis.

The processes used to prepare the video files for processing with NGSIM-VIDEO are described in the subsequent sections.

Figure 6.7 Study Area and Camera Coverage

Figure 6.8 Study Area Schematic



Video Stabilization

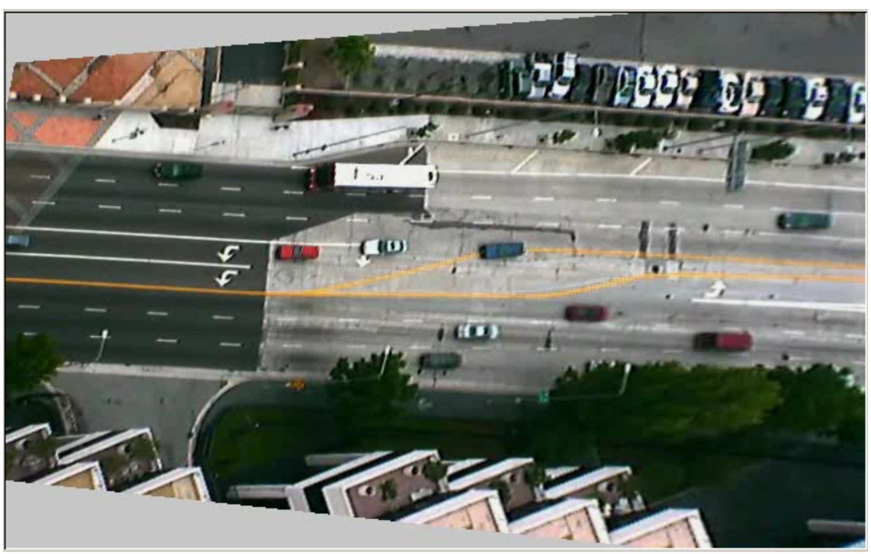
An assessment of the video files determined that stabilization was not required, and therefore not performed, for any of the collected video.

Video Rectification

All five cameras were calibrated using the same settings recorded during field video collection. All video files were rectified by following the procedures discussed in Section 3.0. Figure 6.9 shows a snapshot of the source AVI from Camera 3. The corresponding rectified image is shown in Figure 6.10.

Figure 6.9 Source AVI (Camera 3)

Figure 6.10 Rectified AVI (Camera 3)



Configuration File

As discussed in Section 3.0, the configuration file is in XML format and includes AVI file names and locations, camera's intrinsic parameters and transformation matrices, input parameters, zoning, and direction polygons with directional points. An overview of the configuration file is illustrated in Figure 6.11. To review the contents of the complete configuration file, please browse through the file which is included in the NGSIM-VIDEO installation package.

Figure 6.11 Configuration File for Lankershim Boulevard


```
<?xml version="1.0" encoding="UTF-8" ?>
- <ProcessInput xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="D:\Work1\Ngsim\data-extraction2\data-
    model\process-input11.xsd">
    <xInputUnitsToMeters>0.3048</xInputUnitsToMeters>
    <InputUnitsToMeters>1.0</InputUnitsToMeters>
    <MaximumVelocity>8.91,-8.04,0.0</MaximumVelocity>
    <DefaultVelocity>5.94, -5.36, 0.0</DefaultVelocity>
    <StandardVehicleWidth>7.0</StandardVehicleWidth>
  + <Cameras ViewPoint="1" Id="1" Description="C:\Lankershim_AM_Jun16_05
      \AVI_0845To0900
      \LA_Lankershim_Jun16_2005_AM_Cam1_0845To0900.avi">
  + <Cameras ViewPoint="1" Id="2" Description="C:\Lankershim_AM_Jun16_05
      \AVI_0845To0900
      \LA_Lankershim_Jun16_2005_AM_Cam2_0845To0900.avi">
  + <Cameras ViewPoint="1" Id="3" Description="C:\Lankershim_AM_Jun16_05
      \AVI_0845To0900
      \LA_Lankershim_Jun16_2005_AM_Cam3_0845To0900.avi">
  + <Cameras ViewPoint="1" Id="4" Description="C:\Lankershim_AM_Jun16_05
      \AVI_0845To0900
      \LA_Lankershim_Jun16_2005_AM_Cam4_0845To0900.avi">
  + <Cameras ViewPoint="1" Id="5" Description="C:\Lankershim_AM_Jun16_05
      \AVI_0845To0900
      \LA_Lankershim_Jun16_2005_AM_Cam5_0845To0900.avi">
  + <Zoning>
  + <DirectionPolygonVector Id="2430">
  + <DirectionPolygonVector Id="2410">
  + <DirectionPolygonVector Id="2420">
  + <DirectionPolygonVector Id="1430">
  + <DirectionPolygonVector Id="1420">
  + <DirectionPolygonVector Id="1421">
  + <DirectionPolygonVector Id="9120">
  + <DirectionPolygonVector Id="9121">
  + <DirectionPolygonVector Id="9210">
  - <DirectionPolygonVector Id="2130">
My Computer
```


NGSIM-VIDEO Processing

Once the configuration file is loaded and the NGSIM-VIDEO database is connected, NGSIM-VIDEO will launch four windows as discussed in Section 4.0: 1) main control panel, 2) view, 3) alert list, and 4) alert editor windows. Vehicles were detected either automatically or manually when the vehicle first appeared in the study area either on Lankershim Boulevard or on side streets. In a total, there were 11 detection zones (in blue color in Figure 6.12) and 10 drop zones (in yellow color in Figure 6.12).

Tracking was not separated for forward and reverse processing. In other words, all five cameras were included in one tracking view window, as illustrated in Figure 6.12. Following the tracking procedure described in earlier sections resulted in successful application of NGSIM-VIDEO, as illustrated in Figure 6.13 which shows tracking boxes with VIDs on top of the vehicles.

Figure 6.12 NGSIM-VIDEO View Window for Lankershim Data (Partial View)

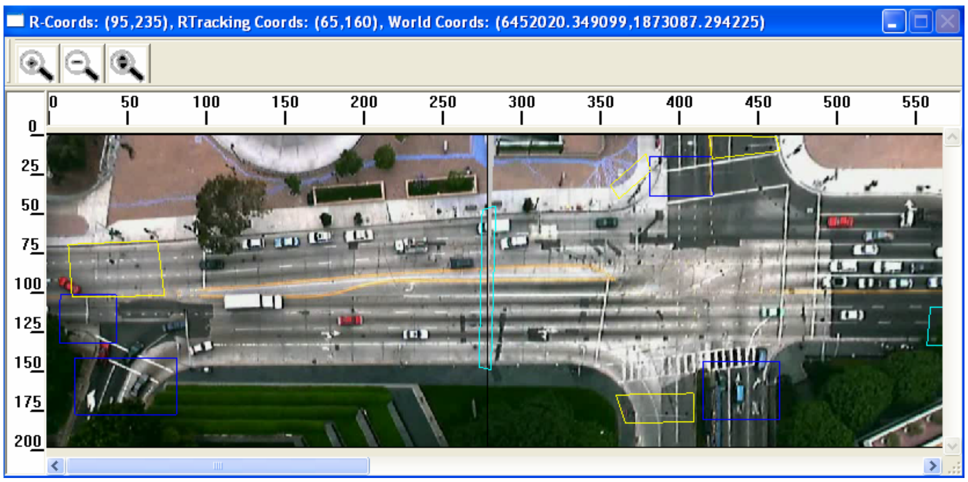
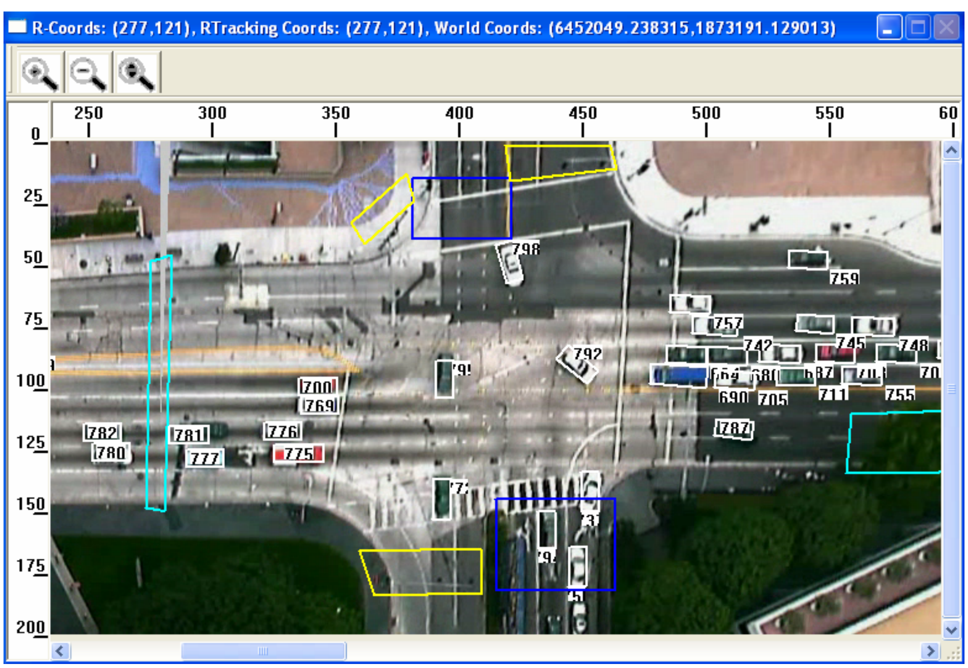


Figure 6.13 NGSIM-VIDEO View Window with Tracking Boxes for Lankershim Data



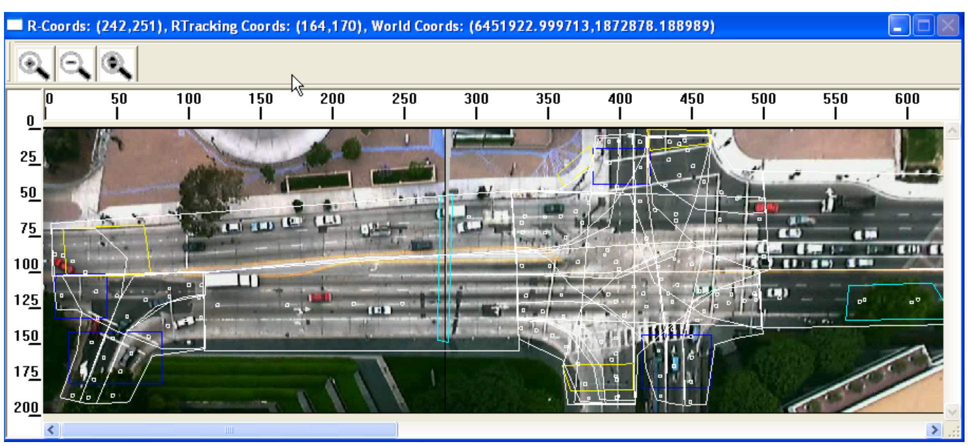
NGSIM-VIDEO Database

The results of the NGSIM-VIDEO processing effort are useful arterial dataset. To provide an example of the magnitude of this dataset, a total number of 1,231 vehicles, representing about one million trajectory data points, were tracked for the time period of 8:45 a.m. to 9:00 a.m.

Notes

- Considering different vehicle moving orientations for arterial tracking, it is recommended to turn off automatic detection and use manual detection instead. This requires more manual tracking effort, but it helps generate more appropriate tracking boxes with correct sizes for every vehicle. To do this, the user should left click on the upper left (UL, as defined in Figure 5.2) of the vehicle once, and then move the arrow to the upper right (UR, as defined in Figure 5.2) of the vehicle and click again. These two clicks together with the default vehicle width (as defined in the configuration file) create a rectangle over the vehicle and tells NGSIM-VIDEO where to look for a vehicle. It also assigns a VID to that vehicle in sequential order.
- Direction polygons are required for arterial tracking. Unlike freeways, arterials represent more complicated turning movements. For each different movement, it is necessary to create a direction polygon as well as associated directional points. For Lankershim Boulevard tracking, about 60 direction polygons were created. Figure 6.14 illustrates part of the direction polygons as well as associated directional points inside the polygons (in white color). In order to display those directional polygons in the View window, the user must select "Display Motion Zones" in Options menu in the Main Control Panel.

Figure 6.14 Direction Polygons and Points



A. Visual Basic Code for Getting Pixel Coordinates of Polylines and Polygons in ArcGIS

Figure A.1 Visual Basic Code for Getting Pixel Coordinates of Polylines and Polygons in ArcGIS

```
'Change shapefile name and output file name here;
'then click the right arrow to run the program;
'Remember to put the cursor to this main() function.

Private Sub MAIN()
   FeatureVertices2Text "camera-location," "allCam_edgePoints.txt"
End Sub
Public Function FeatureVertices2Text(sLayerName As String,
sTextFileName As String, _
Optional nMaxFeatures As Double)
' loop through layer features, up to nMaxFeatures
' write out x, y, z, and m values to sTextFileName, as well as
summary info

Dim pFeat As IFeature
Dim pLayer As ILayer
Dim pMxDoc As IMxDocument
Dim pEnumLayers As IEnumLayer
Dim i As Integer
Dim pFeatClass As IFeatureClass
Dim pFeatLayer As IFeatureLayer
Dim pFeatCursor As IFeatureCursor
Dim sZ As String, sY As String, sX As String, sM As String, sFeatN
As String, sFeatVerts As String
Dim lFileID As Long
Dim pFeatPoints As IPointCollection
Dim pPoints As Point
Dim nFeat As Double
Dim nX, nY, nZ, nM

Set pMxDoc = Application.Document
Set pEnumLayers = pMxDoc.FocusMap.Layers

' find the requested layer:
  Set pLayer = pEnumLayers.Next
  Do While Not pLayer Is Nothing
    If UCase(pLayer.Name) = UCase(sLayerName) Then Exit Do
    Set pLayer = pEnumLayers.Next
  Loop

  If pLayer Is Nothing Then
     Exit Function
  End If
```

```
' get the feature cursor:
  Set pFeatLayer = pLayer
  Set pFeatClass = pFeatLayer.FeatureClass
  Set pFeatCursor = pFeatClass.Search(Nothing, False)
' open file:
  lFileID = FreeFile()
  If Len(Dir(sTextFileName)) > 0 Then Kill sTextFileName
  Open sTextFileName For Append As lFileID
' loop through the features:
  Set pFeat = pFeatCursor.NextFeature
  Print #lFileID, "Id" & vbTab & "VerNum" & vbTab & "XCoor" & vbTab &
"YCoor"

  Do While Not pFeat Is Nothing
      nFeat = pFeat.Value(2)
      'If (nFeat > nMaxFeatures) And (nMaxFeatures > 0) Then
      '  Exit Do
      'End If

  ' loop through the vertices of the features:
      Set pFeatPoints = pFeat.Shape
      For i = 0 To pFeatPoints.PointCount - 1
          sFeatN = nFeat & vbTab
          sFeatN = sFeatN & i + 1 & vbTab
          nX = pFeatPoints.Point(i).X
          nY = pFeatPoints.Point(i).Y
          sX = nX & vbTab
          sY = nY
          sFeatVerts = sFeatN & sX & sY
          Print #lFileID, sFeatVerts
      Next
      Set pFeat = pFeatCursor.NextFeature
  Loop
  Close lFileID
End Function
```

U.S. Department of Transportation
400 Seventh Street S.W., Room 4410
Washington, DC 20590
http://www.fhwa.dot.gov

Publication No.: FHWA-HOP-07-009

Publication Date: March 2007